Universal Connector

User Guide

Indesca / Infitran

Version 4.1.0

usap-user-4100

Universal Connector

User Guide

Indesca / Infitran 4.1.0

Document Name	Universal Connector 4.1.0 User Guide				
Document ID	usap-user-4100				
Products	z/OS	UNIX	Windows	OS/400	HP NonStop
Universal Connector	√	√			

Stonebranch Documentation Policy

Stonebranch, Inc.
950 North Point Parkway, Suite 200
Alpharetta, GA 30005 USA
Tel: (678) 366-7887
Fax: (678) 366-7717

SAP Certified Integration

Summary of Changes

Changes for Universal Connector 4.1.0 User Guide (usap-user-4100)
February 10, 2010

Universal Connector 4.1.0.0

- Added Section 1.2.2 XBP 3.0 Support.
- Added Section 2.7 Mass Activities Support.
- Added the following in Chapter 3 Universal Connector for z/OS:
  - Section 3.3.13 MASS ACTIVITY WAIT Command.
  - RETURN_APPLICATION_LOG (-applog), RETURN_APPLICATION_RC (-printapprc), and USE_APPLICATION_RC (-useapprc) configuration options to sections:
    - 3.3.12 WAIT for JOB Command
    - 3.3.15 BDCWAIT Command
- Added the following in Chapter 4 Universal Connector for UNIX:
  - Section 4.3.13 MASS ACTIVITY WAIT Command.
  - RETURN_APPLICATION_LOG (-applog), RETURN_APPLICATION_RC (-printapprc), and USE_APPLICATION_RC (-useapprc) configuration options to sections:
    - 4.3.12 WAIT for JOB Command
    - 4.3.15 BDCWAIT Command

Changes for Universal Controller 3.2.0 User Guide (usap-user-3203) September 8, 2009

Universal Connector 3.2.0.2

- Added SPOOL_CODEPAGE configuration option for the following commands in Chapter 3 Universal Connector for z/OS:
  - RUN JOB Command
  - WAIT for JOB Command
  - DISPLAY SPOOLLIST Command
- Added SPOOL_CODEPAGE configuration option for the following commands in Chapter 4 Universal Connector for UNIX:
  - RUN JOB Command
  - WAIT for JOB Command
  - DISPLAY SPOOLLIST Command

Changes for Universal Connector 3.2.0 User Guide (usap-user-3202) April 1, 2009

Universal Connector 3.2.0.1

- Added Section 2.6 z/OS CANCEL Command Support.
- Added SERVER_STOP_CONDITIONS option for the folowing z/OS commands:
  - RUN JOB Command
  - WAIT for JOB Command
- Added TARGET_VARIANT option for the following z/OS commands:
  - RUN JOB Command
  - SUBMIT JOB Command
- Added TARGET_VARIANT option for the following UNIX commands:
  - RUN JOB Command
  - SUBMIT JOB Command
- Added INSTALLATION Options in Chapter 4 Universal Connector for UNIX.

Changes for Universal Connector 3.2.0 User Guide (usap-user-3201) September 5, 2008

- Added toll-free telephone number for North America in Appendix A Customer Support.

Changes for Universal Connector 3.2.0 User Guide (usap-user-320) May 16, 2008

Universal Connector 3.2.0.0

- Added the following command to Chapter 3 Universal Connector for z/OS:
  - DISPLAY INTERCEPTED_JOBS Command
- Added the following command to Chapter 4 Universal Connector for UNIX:
  - DISPLAY INTERCEPTED_JOBS Command
- Added options to the following sections:
  - Chapter 3 Universal Connector for z/OS
    - DISPLAY VARIANT Command
    - HOST Options
    - MESSAGE Options
  - Chapter 4 Universal Connector for UNIX
    - DISPLAY VARIANT Command
    - HOST Options
    - MESSAGE Options
- Added Section 3.4.6 EVENT Options to Chapter 3 Universal Connector for z/OS.
- Added Section 4.4.6 EVENT Options to Chapter 4 Universal Connector for UNIX.
- Specified requirement for XBP interface 2.0 for the following commands:
  - Chapter 3 Universal Connector for z/OS
    - SUBMIT VARIANT Command
    - MODIFY VARIANT Command
    - DISPLAY VARIANT Command
    - GENERATE VARIANT DEFINITION FILE Command
  - Chapter 4 Universal Connector for UNIX
    - SUBMIT VARIANT Command
    - MODIFY VARIANT Command
    - DISPLAY VARIANT Command
    - GENERATE VARIANT DEFINITION FILE Command
- Removed RETURN_CODE_PAIR options from the following commands:
  - Chapter 3 Universal Connector for z/OS
    - RUN FS JOB NETWORK Command
    - START FS JOBNET Command
    - WAIT for FS JOB NETWORK Command
  - Chapter 4 Universal Connector for UNIX
    - RUN FS JOB NETWORK Command
    - START FS JOBNET Command
    - WAIT for FS JOB NETWORK Command

Changes for Universal Connector for SAP 3.1.1 User Guide (usap-user-31111) February 28, 2007

- Added List of Figures and List of Tables.
- Added Appendix A Customer Support.
- Divided Section 2 into multiple, platform-specific sections.

Universal Connector for SAP 3.1.1.2

- Added MAX_JOB_LOG_SIZE and MAX_SPOOL_LIST_SIZE options to the following sections of Section 3 UNIX and Section 4 z/OS:
  - 3.1.4.1 RUN JOB Command Options and Figure 2. RUN JOB Command Syntax
  - 3.1.13.1 WAIT for JOB Command Options and Figure 12. WAIT for JOB Command Syntax
  - 3.1.14.1 WAIT for FS JOB NETWORK Command Options and Figure 13. WAIT for FS JOB NETWORK Command Syntax
  - 3.1.21.1 DISPLAY JOBLOG Command Options (MAX_JOB_LOG_SIZE only) and Figure 20. DISPLAY JOBLOG Command Syntax
  - 3.1.22.1 DISPLAY SPOOLLIST Command Options (MAX_SPOOL_LIST_SIZE only) and Figure 21. DISPLAY SPOOLLIST Command Syntax
  - 4.4.4.1 RUN JOB Command Options and Figure 49. RUN JOB Command Syntax
  - 4.4.14.1 WAIT for JOB Command Options and Figure 59. WAIT for JOB Command Syntax
  - 4.4.15.1 WAIT for FS JOB NETWORK Command Options and Figure 60. WAIT for FS JOB NETWORK Command Syntax
  - 4.4.22.1 DISPLAY JOBLOG Command Options (MAX_JOB_LOG_SIZE only) and Figure 67. DISPLAY JOBLOG Command Syntax
  - 4.4.23.1 DISPLAY SPOOLLIST Command Options (MAX_SPOOL_LIST_SIZE only) and Figure 68. DISPLAY SPOOLLIST Command Syntax
- Added error value to PRINT_JOB_LOG_CHILD option in the following sections of Section 3 UNIX:
  - 3.1.6.1 SUBMIT JOB Command Options and Figure 4. SUBMIT JOB Command Syntax
  - 3.1.10.1 MODIFY JOB Command Options and Figure 8. MODIFY JOB Command Syntax
  - 3.1.11.3 START JOB Command Options and Figure 10. START JOB Command Syntax
  - 3.1.13.1 WAIT for JOB Command Options and Figure 12. WAIT for JOB Command Syntax
- Added MAX_CHILD_DEPTH option to the following sections of Section 3 UNIX:
  - 3.1.6.1 SUBMIT JOB Command Options and Figure 4. SUBMIT JOB Command Syntax
  - 3.1.13.1 WAIT for JOB Command Options and Figure 12. WAIT for JOB Command Syntax
- Added error value to Configuration File Keyword of PRINT_JOB_LOG_CHILD option in the following sections of Section 4 z/OS:

  - 4.4.6.1 SUBMIT JOB Command Options
  - 4.4.10.1 MODIFY JOB Command Options
  - 4.4.12.1 START JOB Command Options
  - 4.4.14.1 WAIT for JOB Command Options
- Added -waitchild, -joblogchild, -spoollistchild, and -purgechild options to Figure 57. START JOB Command Syntax in Section 4 z/OS.

Changes for 3.1.1 MVS PTF
January 30, 2006

- Added error value to joblogchild option.
- Added max_child_depth option.
- Removed Windows section.

Changes for 3.1.1 Release
April 30, 2005

- Added configuration option on z/OS platform to specify the target directory for RFC trace file operations.

Documentation Update
February 8, 2005

- Added documentation for new mode of client Fault tolerance (Secure CFT).
- Added discussion of XBP 2.0 support.
- Added section on SAP R/3® user authorization requirements.

Changes for 3.1.0 Release
August 31, 2004

- Support for SAP XBP 2.0 interface.
- Support for XBP parent/child functionality.
- Wait for child jobs.
- Print job log for child jobs.
- Print spoollists for child jobs.
- Support for External Command job steps.
- Purge child jobs.
- Raise SAP events externally.
- Append/replace SAP job intercept criteria table.
- Return SAP syslog for all RFC/BAPI errors.
- Addition of DISPLAY REPORTS command.
- Addition of DISPLAY COMMANDS command.
- Addition of DISPLAY OUTPUT_DEVICES command.

- Addition of DISPLAY PRINT_FORMATS command.

Changes for 1.2.0 Release
January 20, 2004

- Support for SAP R/3® dynamic variants.
- Addition of support for all fields of an SAP R/3® job definition.

Changes for 1.2.0 Release
June 23, 2003

- New z/OS SMP/E installation job condition codes are possible as a result of possible PTF HOLD's.
- New ability to specify the job class when defining an SAP R/3® job with USAP.
- Circumvention for an SAP R/3® condition in which job status is not reported as complete unless a BAPI_XBP_JOB_STATUS_CHECK is performed.

Changes for 1.2.0 Release
March 7, 2003

- Support for Communication Management interface.
- Support for batch input monitoring.
- Express/OS packaging for z/OS installation.

Changes for 1.2.0 Release
August 28, 2002

- New option to specify SAP logon language to be used for USAP session.

Changes for 1.2.0 Release
February 19, 2002

- New option to automatically return the SAP R/3® syslog if a job does not complete successfully.

Changes for 1.2.0 Release
January 9, 2002

- Addition of page limit parameter for DISPLAY syslog command.
- Addition of target jobname parameter to specify a new job name when copying a predefined SAP R/3® job.

Changes for 1.2.0 Release
November 1, 2001

- Universal Connector for SAP for OS/390.
- Ability to format returned SAP spoollists using a translation table.
- Universal Encrypt 2.1.0.
- General documentation improvements.

Changes for 1.2.0 Release
August 2, 2001

- Addition of RFC fault tolerance configuration parameters.
- Addition of WAIT command to reconnect to started jobs.
- Addition of user defined exit codes for SAP R/3® job status.

Changes for 1.2.0 Release
May 11, 2001

- Addition of BAPIXMJOB fields for -output formatting option.
- General documentation improvements.

Changes for 1.2.0 Release
April 3, 2001

- Addition of DISPLAY QState command.

Changes for 1.2.0 Release
January, 2001

- SAP Certification for BC-XBP interface for SAP R/3® release 4.5 and above.
- Addition of the "start job immediately" flag.
- Addition of "target server" parameter for START and RUN commands.
- Addition of the MODIFY command.
- Addition of new query parameters to the DISPLAY SELECT command.
- Addition of output control parameters to the DISPLAY SELECT command.
- Addition of the DISPLAY Variants command.
- Addition of the DISPLAY Syslog command.
- Addition of the set XMI audit level parameter.
- Addition of encrypted command file.
- Addition of the SYNTAX command.

Contents

List of Figures

List of Tables

Preface

Document Structure

This document is written using specific conventions for text formatting and according to a specific document structure in order to make it as useful as possible for the largest audience. The following sections describe the document formatting conventions and organization.

Cross-Reference Links

This document contains cross-reference links to information in its companion document, the Universal Connector Reference Guide.

In order for the links to work correctly:

- Place the documents in the same folder.
- In Adobe Reader / Adobe Acrobat, de-select **Open cross-document link in same window** in the **General** category of your **Preferences** dialog (selected from the **Edit** menu).

Conventions

The following text formatting conventions are used within this document to represent different information.

Typeface and Fonts

`This Font` identifies specific names of different types of information, such as file names or directories (for example, **`\abc\123\help.txt`**).

Command Syntax Diagrams

Command syntax diagrams use the following conventions:

Convention	Description
`bold monospace font`	Specifies arguments and/or values to be typed verbatim, such as file / data set names.
`italic monospace font`	Specifies values to be supplied by the user.
[ ]	Encloses options or values that are optional.
{ }	Encloses options or values of which one must be chosen.
|	Separates a list of possible choices.
. . .	Specifies that the previous item may be repeated one or more times.
BOLD UPPER CASE	Specifies a group of options or values that are defined elsewhere.

Table P.1 Command Syntax

Operating System-Specific Text

Most of this document describes the product in the context of all supported operating systems. At times, it is necessary to refer to operating system-specific information. This information is introduced with a special header, which is followed by the operating system-specific text in a different font size from the normal text.

z/OS

This text pertains specifically to the z/OS line of operating systems.

This text resumes the information pertaining to all operating systems.

Tips from the Stoneman

Stoneman's Tip

Look to the Stoneman for suggestions
or for any other information
that requires special attention.

Vendor References

References are made throughout this document to a variety of vendor operating systems. We attempt to use the most current product names when referencing vendor software. The following names are used within this document:

- **z/OS** is synonymous with IBM z/OS and IBM OS/390 line of operating systems.
- **Windows** is synonymous with Microsoft's Windows 2000 / 2003 / 2008, Windows XP, Windows Vista, and Windows 7 lines of operating systems. Any differences between the different systems will be noted.
- **UNIX** is synonymous with operating systems based on AT&T and BSD origins and the Linux operating system.

These names do not imply software support in any manner. Refer to the Universal Products 4.1.0 Installation Guide for a detailed list of supported operating systems.

Document Organization

The document is organized into the following chapters:

- Overview (Chapter 1)
  General architectural and functional overview of Universal Connector.
- Features (Chapter 2)
  Information about the features and functionality of Universal Connector.
- Universal Connector for z/OS (Chapter 3)
  Information about Universal Connector, specific to the z/OS operating system.
- Universal Connector for UNIX (Chapter 4)
  Information about Universal Connector, specific to the UNIX operating system.
- Job Definition Files (Chapter 5)
  Information about the Universal Connector job definition files, which contain statements that specify the attributes of jobs
- Customer Support (Appendix A)
  Customer support contact information for Universal Connector.

Chapter 1
Overview

1.1 Introduction to Universal Connector

Universal Connector is a command line application that controls background processing within an SAP system. This allows any computer on the network to manage SAP background processing tasks via the local command line.

You indicate to Universal Connector which SAP system to connect to and what background processing tasks to perform. Universal Connector connects to the SAP system and processes your request.

On z/OS and UNIX, Universal Connector is part of Universal Products, which provides command line interfaces to all of the major operating systems in your data center. That is, the remote operating system's command line interface is extended to the local operating system's command line interface. The remote and local systems can be running two different operating systems.

All of the Universal Products components can interact with Universal Connector.

1.1.1 Universal Connector Functionality

Universal Connector provides the functionality to integrate SAP systems into both local administrative tools and enterprise system management infrastructures.

Specifically, Universal Connector allows you to:

- Define SAP jobs using a job definition file or by copying existing jobs.
- Modify SAP jobs using a job definition file.
- Start SAP jobs.
- Check the status of SAP jobs.
- Retrieve the joblog of SAP jobs.
- Retrieve the spoollists of SAP jobs.
- Delete SAP jobs and their associated output.
- Query jobs in the SAP system.
- Define SAP variants using a variant definition file.
- Modify SAP variants using a variant definition file.
- Query variants in the SAP system.
- Process/monitor Batch Input sessions.
- Retrieve the SAP syslog.
- Define SAP FS job networks to the SAP system using a definition file.
- Start SAP FS job networks.
- Check the status of SAP FS job networks.
- Delete SAP FS job networks from the SAP system.
- Interface with the MHP Communication Management product.

1.1.2 Universal Connector Communications

Universal Connector communicates with an SAP system using an SAP RFC connection. Through this RFC connection, Universal Connector utilizes SAP's external interfaces to perform background-processing tasks.

1.2 Supported SAP Versions

Universal Connector supports SAP 3.1G and above.

The following commands are not available when running USAP against SAP 3.1 and 4.0 systems:

- Purge.
- Display select.
- Target Server parameter for START and RUN commands.

1.2.1 XBP 2.0 Support

Universal Connector supports the SAP XBP 2.0 interface. The XBP 2.0 interface introduces important new feature sets and many enhancements to basic functionality.

The following features are the most notable additions to the XBP 2.0 interface:

Parent / Child Functionality

This feature allows Universal Connector to identify the parent/child relationship between jobs and work with them accordingly. For example, monitoring a submitted job can now take into account the activity of all child jobs.

Job Intercept Functionality

This feature allows Universal Connector to define and modify criteria used by the SAP system to intercept jobs (prevent jobs from starting).

Raise Events Externally

This feature allows Universal Connector to trigger SAP events.

IMPORTANT: Some features of the XBP 2.0 interface (parent/child and interception) may not be used by all SAP customers. Therefore, to prevent unnecessary use of resources, SAP provides a means to globally turn on and off these features. ABAP program **`INITXBP2`** performs this function.

Parent/child functionality and job interception functionality are turned off by default. The SAP ABAP program **`INITXBP2`** must be run before Universal Connector can use this functionality.

1.2.2 XBP 3.0 Support

USAP supports the SAP XBP 3.0 interface. All functionality will go through the XBP 3.0 interface if it is available.

Currently, Universal Connector supports only the following new feature set of XBP 3.0:

Application Information

This feature set includes the ability to retrieve application logs and application return codes for jobs on the SAP system.

Note: Not all jobs will create this information. The availability is dependent upon the functionality of the programs that are executed within the job on the SAP system.

1.3 SAP User Authorization Requirements

USAP requires a user ID defined in the SAP system for RFC logon/user authentication. The user ID used with USAP requires certain SAP authorizations to perform tasks within the SAP system.

If the instance profile parameter **auth/rfc_authority_check** is set to 1, the system checks authorization for the function group of the RFC function module against the authorization object **S_RFC**. In this case, the following authorizations are required:

SAP 3.1 - 4.0

User IDs that will run USAP should be assigned an authorization for the authorization object **S_RFC** with the following fields:

- Type of RFC object to be protected (RFC_TYPE)=FUGR.
- Name of the RFC object (RFC_NAME)=SXMI, SXJI, SQUE, STUS.
- Activity (RFC_ACTVT)=16 (execute).

SAP 4.5 (and Higher)

User IDs that will run USAP should be assigned an authorization for the authorization object **S_RFC** with the following fields:

- Type of RFC object to be protected (RFC_TYPE)=FUGR.
- Name of the RFC object (RFC_NAME)=SXMI, SXBP, SQUE, STUS.
- Activity (RFC_ACTVT)=16 (execute).

Chapter 2
Features

2.1 Overview

This chapter provides information on Universal Connector features that apply to all operating systems.

- Configuration
- Remote Configuration
- Network Fault Tolerance
- Client Fault Tolerance
- CM Interface
- Batch Input Monitoring
- Message and Audit Facilities

2.2 Configuration

Product configuration consists of:

- Setting default options and preferences for all executions of Universal Connector.
- Setting options and preferences for a single execution of Universal Connector.

Options control product behavior and resource allocation.

- An example of configurable product behavior is whether or not data transferred over the network is compressed.
- An example of configurable resource allocation is the directory location in which the product creates its log files.

Each option is comprised of a pre-defined parameter, which identifies the option, and one or more values. The format of the parameter depends on the method being used to specify the option.

Although there are many configurable product options, Universal Products, in general, are designed to require minimal configuration and administration. The default options will work very well in most environments. When local requirements do require a change in product configuration, there are multiple methods available to configure the products in order to meet your needs.

2.2.1 Configuration Methods

All Stonebranch Inc. Universal Products provide a consistent and flexible method of configuration. An operating system's native configuration methods, such as configuration files, are utilized in order to integrate with existing system management policies and procedures for the platform.

Depending on specific Universal Products, and the operating system on which it is being run, product configuration is performed by one or more methods. These configuration methods, in their order of precedence, are:

1. Command Line
2. Command Line File
3. Environment Variables
4. Configuration File

This order of precedence means that an option specified on the command line overrides the same option specified in a command file, which overrides the same option specified with an environment variable, which overrides the same option specified in a configuration file

Note: For security reasons, not all options can be overridden.

2.2.2 Command Line

Configuration options specified on a command line affect one instance of a program execution. Each time that you execute a program, command line options let you tailor the behavior of the program to meet the specific needs for that execution.

Command line options are the highest in order of precedence of all the configuration methods (see Section 2.2.1 Configuration Methods). They override the options specified using all other configuration methods, except where indicated.

Command line options consist of:

- Parameter (name of the option)
- Value (pre-defined or user-defined value of the option)

The command line syntax depends, in part, on the operating system, as noted below.

An value may or may not be case-sensitive, depending on what it is specifying. For example, if a value is either **yes** or **no**, it is not case-sensitive. It could be specified as **YES**, **Yes**, or **yes**. However, if a value specifies a directory name or file name, it would be case-sensitive if the operating system's file system is case-sensitive.

If an option is specified more than once on the command line, the last instance of the option specified is used.

z/OS

z/OS command line options are specified in the JCL EXEC statement PARM keyword or on the SYSIN ddname. The PARM keyword is used to pass command line options to the program being executed with the EXEC statement.

Command line options are prefixed with a dash (**-**) character. For many options, there are two different forms in which they can be specified:

- Short form: one case-sensitive character
- Long form: two or more case-insensitive characters

The parameter and value must be separated by at least one space.

Example command line options specified in the PARM value follow:

Short form:

```
PARM='-l INFO -G yes'
```

Long form:

```
PARM='-LEVEL INFO -LOGIN YES'
```

As noted above, z/OS command line options also can be specified on the SYSIN ddname. This is the easiest and least restrictive place to specify options, since the PARM values are limited in length. The options specified in the SYSIN ddname have the same syntax. Options can be specified on one line or multiple lines. The data set or inline data allocated to the SYSIN ddname cannot have line numbers in the last 8 columns (that is, all columns of the records are used as input).

All columns of the SYSIN data set are read. Sequence numbers at the end of the record will be read as data if present and terminate the batch job with an error. Therefore, remove all sequence numbers from the SYSIN data set. From within the ISPF editor, sequence numbers are removed with the UNNUMBER primary command.

UNIX

UNIX command line options are prefixed with a dash (**-**) character.

For many options, there are two different forms in which they can be specified:

- Short form: one case-sensitive character.
- Long form: two or more case insensitive characters.

The parameter and value must be separated by at least one space or tab character.

Example command line options follow:

Short form:

```
-l info -G yes
```

Long form:

```
-level info -login yes
-LEVEL info -LoGiN YES
```

2.2.3 Command Line File

A command line file contains command line options specified in a file. The command line file enables you to save common command line options in permanent storage and reference them as needed.

The command line file is the second to highest in the precedence order after command line options (see Section 2.2.1 Configuration Methods).

Individual command line options can be specified on one or multiple lines. Blank lines are ignored. Lines starting with the hash (**#**) character are ignored and can be used for comments.

The command line file can be encrypted if it is necessary to secure the contents.

Note: If the contents of the file contain sensitive material, the operating system's native file and user security facilities should be used in addition to the file encryption provided by the Universal Products.

2.2.4 Environment Variables

Environment variables, like command line options, allow options to be specified for one instance of a program execution. Each time that you execute a program, environment variables allow you to tailor the behavior of the program to meet the specific needs for that execution.

Environment variables are the third to highest in the precedence order after command line file options (see Section 2.2.1 Configuration Methods).

Each operating system has its own unique method of setting environment variables.

All environment variables used by Universal Products are upper case and are prefixed with a product identifier consisting of three or four characters. The product sections specify the value of the environment variables. Values are case-sensitive.

z/OS

Environment variables are specified in the JCL EXEC statement PARM keyword. Environment variables are part of the IBM Language Environment (LE) and as such are specified as LE runtime options. The PARM value is divided into LE options and application options by a slash (/) character. Options to the left of the slash are LE options and options to the right are application options.

Example of setting an environment variable:

Set option UCMDLEVEL to a value of INFO:

```
PARM='ENVAR("UCMDLEVEL=INFO")/'
```

UNIX

Environment variables in UNIX are defined as part of the shell environment. As such, shell commands are used to set environment variables. The environment variable must be exported to be used be a called program.

Example of setting an environment variable:

Set option UCMDLEVEL to a value of INFO in a bourne, bash, or korn shell:

```
UCMDLEVEL=INFO
export UCMDLEVEL
```

2.2.5 Configuration File

Configuration files are used to specify system-wide configuration values. They provide the simplest method of specifying configuration options whose values will not change with each command invocation.

They are last in precedence order for specifying configuration options (see Section 2.2.1 Configuration Methods). These configuration file "default" values are used if the options are not read from one or more other sources.

Some options only can be specified in the configuration file. Other options cannot be specified in the configuration file; they must be specified via one or more other sources for a single execution of Universal Connector.

If an option is specified more than once in a configuration file, the last option specified is used.

All configuration files on a system are maintained by the local Universal Broker. The Universal Broker serves the configuration data to other Universal Products running on the local system. (The one exception is Universal Enterprise Controller (UEC), which directly reads its own configuration files.

The Universal Broker reads the configuration files when it first starts or when it receives a REFRESH command from Universal Control or Universal Enterprise Controller. Any changes made to a configuration file are not in effect until the Broker is recycled or receives a REFRESH command.

Universal Product components do not read the configuration files themselves. When a component starts, it first registers with the locally running Universal Broker. As part of the registration process, the Broker returns the components configuration data.

When the Universal Broker is operating in managed mode, the configuration information for the various Universal Products is "locked down" and can be modified or viewed only via the I-Management Console (see Section 2.3.2 Managed Mode).

z/OS

Configuration files are members of a PDSE. The data set record format is fixed or fixed block with a record length of 80. No line numbers can exist in columns 72-80. All 80 columns are processed as data.

The USAP configuration file is allocated to ddname **UNVCONF**. This file can be edited manually with the ISPF editor.

See Section 2.2.6 Configuration File Syntax for the configuration file syntax.

UNIX

Configuration files are regular text files on UNIX. The USAP configuration file is named **`usap.conf`**.

Universal Broker searches for the configuration files in a fixed list of directories. The Broker will use the first configuration file that it finds in its search. The directories are listed below in the order they are searched:

Directory	Notes
/etc/opt/universal	
/etc/universal	Installation default
/etc/stonebranch	Obsolete as of version 2.2.0
/etc	
/usr/etc/universal	
/usr/etc/stonebranch	Obsolete as of version 2.2.0
/usr/etc	

Table 2.1 UNIX Configuration File Directory Search

See 2.2.6 Configuration File Syntax for the configuration file syntax.

2.2.6 Configuration File Syntax

Configuration files are text files that can be edited with any available text editor.

The following rules apply for configuration file syntax:

- Options are specified in a keyword / value format.
- Keywords can start in any column.
- Keywords must be separated from values by at least one space or tab character.
- Keywords are not case sensitive.
- Keywords cannot contain spaces or tabs.
- Values can contain spaces and tabs, but if they do, they must be enclosed in single (**'**) or double (**"**) quotation marks. Repeat the enclosing characters to include them as part of the value.
- Values case sensitivity depends on the value being specified. For example:
  - Directory and file names are case sensitive.
  - Pre-defined values (such as **yes** and **no**) are not case sensitive.
- Each keyword / value pair must be on one line.
- Characters after the value are ignored.
- Newline characters are not permitted in a value.
- Values can be continued from one line to the next either by ending the line with a:
  - Plus (**+**) character, to remove all intervening spaces.
  - Minus (**-**) character, to preserve all intervening spaces between the end of the line being continued and the beginning of the continuing line.

  Ensure that the line continuation character is the last character on a line.
- Comment lines start with a hash (**#**) character.
- Blank lines are ignored.

Note: If an option is specified more than once in a configuration file, the last option specified is used.

2.3 Remote Configuration

Universal Products can be configured remotely by Universal Enterprise Controller using the I-Management Console client application, and can be "locked down" so that they *only* can be remotely configured.

I-Management Console instructs the Universal Broker of a remote Agent to modify the configurations of the Universal Products components managed by that Universal Broker.

Universal Broker supports remote configuration in either of two modes:

1. Unmanaged Mode
2. Managed Mode

2.3.1 Unmanaged Mode

Unmanaged mode is the default mode of operations for Universal Broker. It allows a Universal Broker – and the Universal Products components managed by that Universal Broker – to be configured either:

- Locally, by editing configuration files.
- Remotely, via I-Management Console.

The system administrator for the machine on which an Agent resides can use any text editor to modify the configuration files of the various local Universal Products.

Via I-Management Console, selected users can modify all configurations of any Agent, including the local Agent. I-Management Console sends the modified data to the Universal Broker of that agent, which Universal Broker then uses to update the appropriate configuration files.

If I-Management Console sends modifications for a Universal Broker configuration, Universal Broker validates the modified data before it accepts it. If the data fails validation, Universal Broker does not update its configuration file.

If I-Management Console sends modification to the configuration of any other Universal Products component, the Universal Broker updates the appropriate configuration file. The component will use this new configuration at its next invocation.

Note: If errors or invalid configuration values are updated via I-Management Console for a component other than Universal Broker, the component may not run successfully until the configuration has be corrected.

2.3.2 Managed Mode

When a Universal Broker is operating in managed mode, the configuration information for all Universal Products components managed by that Universal Broker is "locked down." Universal Broker stores the information in a database file located within its specified spool directory. The information can be modified only via I-Management Console.

From this point on, Universal Broker uses the database file – not the configuration files – to access configuration information. Any configuration changes made to the components – via I-Management Console – are placed in the database file. Therefore, as long as Universal Broker stays in managed mode, the configuration files may no longer contain current or valid configuration information.

If managed mode is de-selected for the Universal Broker, it reads the database file where it stored the configuration information. Universal Broker uses this information to create and/or update configuration files for the components.

- If a configuration file exists in the configuration directory, it is overwritten.
- If a configuration file does not exist, it is created.

Note: Because of remote configuration and the desire to be able to "lock down" all product configurations, Universal Broker – and all Universal Products servers – no longer support the command line and environmental variables methods of specifying configuration options.

Selecting Managed Mode

The managed mode of operations for Universal Broker is selected via the I-Administrator client application.

(See the Universal Enterprise Controller 4.1.0 Client Applications User Guide for specific information on how to select managed mode.)

Figure 2.1, below, illustrates remote configuration for one Agent in managed mode and one Agent in unmanaged mode.

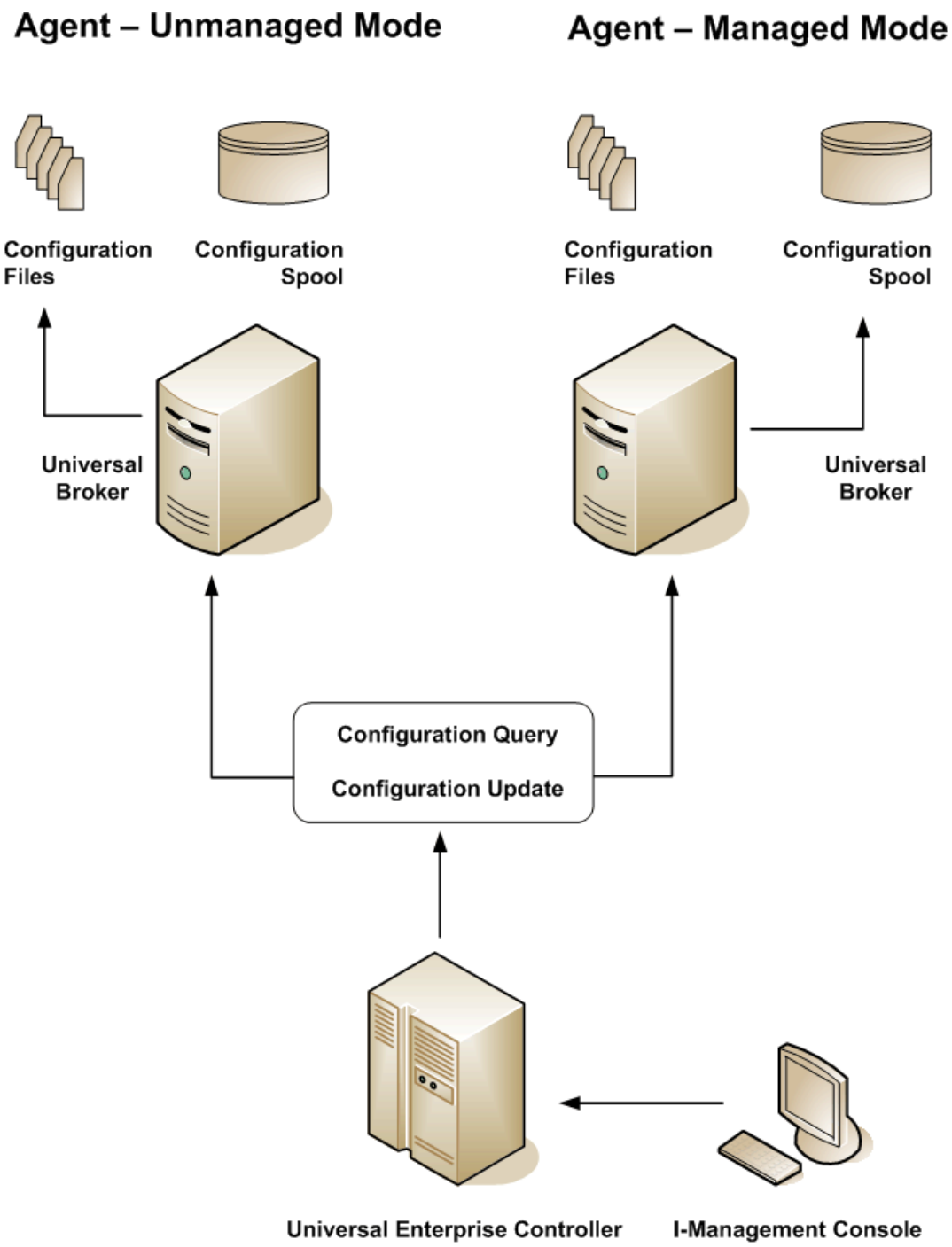

Figure 2.1 Remote Configuration - Unmanaged and Managed Modes of Operation

2.3.3 Universal Broker Start-up

At Universal Broker start-up, in both managed and unmanaged modes, the Universal Broker configuration file is always read.

Unmanaged Mode

At Universal Broker start-up in unmanaged mode, Universal Broker reads the configuration files of all Universal Products components into its memory. The Universal Broker configuration file is used to define the Universal Broker configuration, just as all configuration files are used in unmanaged mode. Universal Broker updates its memory from the configuration files whenever Universal Control issues a REFRESH request.

Managed Mode

At Universal Broker start-up in managed mode, the Universal Broker configuration file points Universal Broker to the location of the configuration spool file, from which the Broker retrieves configuration information for all Universal Products components. Universal Broker updates its memory from the configuration spool file and, automatically, after changes are made via I-Management Console.

If more configuration information than needed is included in the Universal Broker configuration file at Universal Broker start-up, Universal Broker will update its running configuration with the information that it retrieved from the spool file. The configuration file that was used at start-up is made obsolete.

2.4 Network Fault Tolerance

USAP commands are processed by calling appropriate BAPI functions in the SAP system. The BAPI function calls are issued over an RFC connection. USAP provides fault tolerance at the RFC level. If an RFC call fails, that call is retried until it completes successfully, or exceeds a user definable retry limit.

If an RFC call fails, USAP will close the current RFC connection and establish a new RFC connection in order to continue processing. The process of establishing and preparing an RFC connection is referred to in this document as the RFC logon process. The RFC logon process involves establishing an RFC connection, logging on to the XMI interface and setting the XMI audit level. If the RFC logon process fails, it will be retried until it completes successfully, or exceeds a user definable retry limit. When the new RFC connection is successfully established, USAP will reissue the failed RFC call.

The entire process of establishing a new RFC session and reissuing the failed RFC call will be retried until either

- RFC call completes successfully.
- User-definable RFC retry limit is exceeded.

Certain BAPI functions should not be retried in an unknown state. Those BAPI functions are points of failure within the USAP fault tolerant solution. Section 2.4.1 Points of Failure lists the points of failure and their relationship to USAP commands.

2.4.1 Points of Failure

The points of failure within USAP fault tolerant architecture are:

- Job Submission
  Some BAPI functions called in the submission process cannot be restarted in an unknown state without possible negative consequences. If an RFC call fails issuing those BAPI's, USAP will end unsuccessfully.
- Job Modification
  Some BAPI functions called in the job modification process can not be restarted in an unknown state without possible negative consequences. If an RFC call fails issuing those BAPI's, USAP will end unsuccessfully.
- Job Start
  Some BAPI functions called in the job start process can not be restarted in an unknown state. If an RFC call fails issuing those BAPI's, USAP will end unsuccessfully.

2.4.2 Network Fault Tolerance Configuration Parameters

The set of USAP configuration parameters that can be used to fine-tune the fault tolerance support for a particular environment are:

- RFC Logon Retry Count
- RFC Logon Retry Timeout
- RFC Timeout
- RFC Retry Count
- RFC Retry interval
- RFC Listen Interval

See Sections 3.4.4 CFT Options and 3.4.6 EVENT Options for details concerning the use of these parameters.

2.5 Client Fault Tolerance

The Client Fault Tolerance feature allows the USAP client application to be shut down and restarted at a later time. This functionality helps avoid problems that can result if the USAP application terminates unexpectedly while processing an SAP job. In such an instance, the Client Fault Tolerance restart capability allows USAP to reconnect to a running (or completed) job while preventing the unintentional start of a new instance of the original SAP job.

2.5.1 Overview

To achieve Client Fault Tolerance, USAP must be able to associate the SAP jobs it defines and starts with a particular unit of work. In this context, a unit of work includes the USAP client and SAP job instance.

To associate an SAP job with a particular unit of work, the user must be able to specify some identifying characteristic that is specific to that unit of work. The SAP system uniquely identifies job instances by a job name/job ID pair. Since the job name must be reusable and the job ID is assigned by the SAP system at the time of definition, USAP must use an alternative job characteristic. This alternative job characteristic is the USAP command Identifier.

USAP references a particular unit of work by a job name/Command Identifier combination. The USAP command identifier is tied to the SAP job by appending a Command ID Job Step to the SAP job associated with the USAP command instance. The Command ID job step is required for identification purposes only. Therefore, the program used for the Command ID step is intended to add minimum overhead to the job. The Command ID used for the job is included in the definition of the Command ID job step.

2.5.2 Modes

USAP supports two modes of client fault tolerance:

1. Pre-XBP 2.0 Client Fault Tolerance (CFT)
2. Secure Client Fault Tolerance (Secure CFT)

Pre-XBP 2.0 Client Fault Tolerance (CFT)

This mode is the original implementation of client fault tolerance used prior to the release of XBP 2.0. Due to limitations in the XBP 1.0 interface, USAP client fault tolerance on pre-XBP 2.0 SAP systems uses an external program step as the command ID job step. Using an external program step as the command ID job step has the following security and ease of use drawbacks:

- Security drawback
  Using an external program job step requires the SAP user ID to have authority to run external programs. This authority cannot be given lightly for the following reason: When running an external job step, the SAP system first performs an authorization check to see if the user ID has the right to run an external program. If so, the external program is run under the user ID of the user who started the SAP system
- Ease of use drawback
  Using an external program job step requires a target host be specified for the external program to run on. This requires information about the SAP landscape that may not be readily available. Also, this presents the possibility of the USAP job's parameters becoming out of sync with the SAP landscape.

Secure Client Fault Tolerance (Secure CFT)

This mode is an enhancement of the original implementation. The secure CFT mode requires XBP 2.0 to be installed on the SAP side of the USAP connection. In this mode, an ABAP program step is used for the command ID job step. Using an ABAP program step as the Command ID job step eliminates the security and ease of use drawbacks mentioned above for external program job steps.

- Security
  The execution of ABAP programs and the resources required by them are secured by SAP authorization checks.
- Ease of Use
  ABAP program job steps do not require a target host. They run on whichever application server the job runs on. Therefore, there are no target specific parameters required for secure CFT mode.

The mode of client fault tolerance to be used by USAP is determined by the value of the SECURE_CFT option. Valid values for this option are **yes** and **no**:

- **yes** will cause USAP to use secure CFT mode.
- **no** will cause USAP to use the original pre-XBP 2.0 mode of client fault tolerance.

The default value is **yes**.

Both modes of CFT follow the same basic process flow. When USAP is requested to restart a particular command ID job, it queries the SAP system for all jobs with the specified job name. The list of jobs returned by the SAP system is scanned for a job that contains an appropriate Command ID Job Step. If found, USAP will re-connect to the SAP job instance and satisfy the command line requirements.

USAP is capable of restarting a command ID as long as the associated command ID job remains in the SAP system.

2.5.3 Parameters

Client Fault Tolerance Target Host

The client fault tolerance target host parameter is only required for pre-XBP 2.0 client fault tolerance mode. If the secure CFT mode is being used, the client fault tolerance target host parameter is ignored.

As part of an external program command ID job step definition, SAP requires a target host on which to run the external program (echo). USAP provides the client fault tolerance target host parameter to specify the target host for the command ID job step.

The Client Fault Tolerance Target Host is specified with the CFT_TARGET_HOST option.

Client Fault Tolerance Command Prefix

The client fault tolerance command prefix parameter is only required for pre-XBP 2.0 client fault tolerance mode. If the secure CFT mode is being used, the client fault tolerance command prefix parameter is ignored.

The external program command ID job step has the potential to be run on any operating system reachable by the SAP system. The operating system that the Command ID Job Step runs on is that which exists on the host system specified by the Client Fault Tolerance Target Host parameter. Different operating systems may require commands to be called in different ways. Therefore, the Client Fault Tolerance Command Prefix parameter allows the user to specify the prefix necessary to run the echo command on the host system specified by the Client Fault Tolerance Target Host parameter.

For example, to run the echo command on a Windows system, the following command line would be required for an SAP external job step: **`cmd /C echo`**. Therefore, the Client Fault Tolerance Command Prefix for this system would be: **`cmd /C`**.

The Client Fault Tolerance Command Prefix parameter is specified with the CFT_COMMAND_PREFIX option.

Secure Client Fault Tolerance Option

The mode of client fault tolerance to be used by USAP is determined by the value of the SECURE_CFT option. Valid values for this option are *YES* and *NO*:

- *YES* will cause USAP to use secure CFT mode.
- *NO* will cause USAP to use the original pre-XBP 2.0 mode of client fault tolerance.

The default value is YES.

Client Fault Tolerance ABAP Program

The client fault tolerance ABAP program parameter is only required for secure CFT mode. If the pre-XBP 2.0 CFT mode is being used, the client fault tolerance ABAP program parameter is ignored.

The client fault tolerance ABAP program parameter is used to specify the ABAP program to use for the command ID job step. Any ABAP program can be specified. The USAP internal default ABAP program to use is BTCTEST. BTCTEST is a standard SAP ABAP program that should be available on all SAP systems. It does no real processing and can be considered a dummy program that does not interfere with job processing and places little overhead on the system.

2.5.4 Command ID Job Step

USAP creates Command ID jobs by appending a job step to the user's SAP job being defined to the system. This appended job step is the USAP Command ID Job Step.

Pre-XBP 2.0 CFT Mode

In pre-XBP 2.0 CFT mode, the USAP Command ID Job Step executes the external program echo. A string containing the command ID is inserted in the parameter field of the job step. The echo command is lightweight, does not interfere with the original job, and results in the command ID being printed to the joblog.

Secure CFT Mode

In secure CFT mode, the USAP command ID job step executes an ABAP program. The ABAP program defined to the command id step is user configurable with the covetable parameter. Any ABAP program can be specified. The USAP internal default ABAP program to use is BTCTEST. BTCTEST is a standard SAP ABAP program that should be available on all SAP systems. It does no real processing and can be considered a dummy program that does not interfere with job processing and places little overhead on the system.

2.5.5 Command Identifier

USAP requests client fault tolerance by providing a command identifier. The command identifier is specified on the command line with parameter **`-cmdid`**. The command ID/job name pair identifies the unit of work being executed.

The command ID option provides a command identifier that (paired with job name) uniquely identifies the SAP job on the SAP system. When a USAP job is restarted, it must provide the same command ID identifying the SAP job with which it wants to reconnect.

Providing a unique command ID is not trivial. Many USAP clients may be executing on many different hosts, all executing work on the same SAP system. It is possible for a USAP client to start a restartable job from one host, terminate, and restart on a completely different host.

The command ID value can be any text value up to 245 characters in length. In practical terms, the character set and limits on command line length of the USAP host may impose further restrictions on the value.

2.5.6 Requesting Restart

When a restartable USAP command is initiated, it is either an initial instance or a restarted instance of a command ID. The RESTART option is specified on the command line with parameter **`-restart`**. The RESTART option specifies whether the USAP command instance is requesting a restart of a previous command ID or not. Possible RESTART values are YES, NO, or AUTO. The AUTO value specifies that if there is no existing command ID job on the SAP system, consider this USAP execution the first instance. If there is an existing command ID job, consider this a restart of the command ID. The AUTO value permits automatic restart by eliminating the need to modify the RESTART value for the initial instance and restarted instance.

It is important to note that when using the RESTART AUTO option, USAP will not start a new instance of a job on the SAP system if a job matching the job name/command ID exists in the SAP system. USAP will continue to reconnect to the existing SAP job. Without considering the behavior resulting from the use of RESTART AUTO, it may be possible for one to assume that a job has been run multiple times when, in fact, USAP has been reconnecting to the same job instance. Informational messages are printed by USAP to standard error to indicate the reconnected status but, if the message level is not set to info, the messages will not be seen.

Given the possibility for confusion surrounding the use of RESTART AUTO, a parameter has been introduced to control the use of RESTART AUTO. This parameter is described in the following section.

Controlling Auto Restart

If not properly understood, the auto restart behavior described in the previous section can have potentially serious consequences. For this reason, a parameter has been introduced that will allow or disallow the use of auto restart. The parameter is available at the configuration file level and can be overridden at the command line level.

Configuration File Parameter

The configuration file parameter is **`auto_restart_ok`**.

Valid values for **`auto_restart_ok`** are *yes* and *no*:

- *yes* allows the use of auto restart.
- *no* prevents the use of auto restart.

Command File Parameter

The command line parameter **`-autorestartok`** can be used to override the configuration file setting.

Valid values for **`-autorestartok`** are *yes* and *no*:

- *yes* allows the use of auto restart
- *no* prevents the use of auto restart.

2.5.7 Sample Command Lines For Working With Client Fault Tolerance

Working With Job Definition Files

Initial Run of a Command ID Job

The following example will submit, start, and wait for the command ID job defined in job definition file **`jobdef`**. Because the restart option is set to 'no', USAP will scan the SAP system to ensure that a command ID job with the same job name/command ID pair does not already exist on the system.

If a matching command ID job is found on the SAP system, USAP will exit with an error code before performing the job submission.

Pre-XBP 2.0 CFT Mode

usap -user bob -pwd secret -submit jobdef -start -wait -cft_secure_cft no

-cft_target_host pwdf0196 -cft_cmd_prefix "cmd /C" -cmdid 0000000001 -restart no

Note that the Client Fault Tolerance Command ID Prefix is set up for a Windows host. In many user environments, the Client Fault Tolerance Command ID Prefix parameter can be specified in the configuration file and will never need to be specified on the command line. The same may be true for the Client Fault Tolerance Target Host parameter. The secure CFT option would also be set in the configuration file in most cases.

Secure CFT Mode

usap -user bob -pwd secret -submit jobdef -start -wait -cft_secure_cft yes

-cft_abap BTCTEST -cmdid 0000000001 -restart no

Note that in secure CFT mode, the **`cft_secure_cft`** and **`cft_abap`** parameters would most likely be specified in the USAP configuration file.

Restart of a Command ID Job

In the following example, USAP is requested to restart command ID job 0000000001. USAP will first parse the **`jobdef`** file to determine the jobname, and then scan the SAP system for a matching command ID job.

If a matching command ID job is found, USAP reconnects to that job and satisfies the command line requirements. In this case, that means that if the job has not yet been started, it will be started, USAP will wait for the job to complete (if it hasn't already), and the output will be returned.

If no matching command ID job is found, USAP will terminate with an error code. Appropriate informational messages will be printed to standard error.

Pre-XBP 2.0 CFT Mode

usap -user bob -pwd secret -submit jobdef -start -wait -cmdid 0000000001 -restart yes

-cft_secure_cft no

Secure CFT Mode

usap -user bob -pwd secret -submit jobdef -start -wait -cmdid 0000000001 -restart yes

-cft_secure_cft yes

Run a Command ID Job Using Restart AUTO

In the following example, USAP will first scan the SAP system to determine if a matching command ID job exists. If no matching command ID job is found, USAP considers this to be the initial instance of this command ID job and defines the new command ID job to the SAP system. If a matching command ID job is found, USAP reconnects with the existing SAP job.

After determining if the command ID job is initial or a restart, USAP satisfies the command line requirements.

Pre-XBP 2.0 CFT Mode

usap -user bob -pwd secret -submit jobdef -start -wait -cft_target_host pwdf0196

-cft_cmd_prefix "cmd /C" -cmdid 0000000001 -restart auto -cft_secure_cft no

Secure CFT Mode

usap -user bob -pwd secret -submit jobdef -start -wait -cmdid 0000000001 -restart auto

-cft_secure_cft yes

Working With Pre-defined SAP Jobs

Initial Run of a Command ID Job

The following example will submit, start, and wait for the command ID job defined in the pre-existing SAP job with job name 'JOB_A' and job ID 19561301. Because the restart option is set to 'no', USAP will scan the SAP system to ensure that a command ID job with the same job name/command ID pair does not already exist on the system.

If a matching command ID job is found on the SAP system, USAP will exit with an error code before performing the job submission.

Note that the Client Fault Tolerance Command ID Prefix is set up for a Windows host. In many user environments, the Client Fault Tolerance Command ID Prefix parameter can be specified in the configuration file and will never need to be specified on the command line. The same may be true for the Client Fault Tolerance Target Host parameter.

Pre-XBP 2.0 CFT Mode

usap -user bob -pwd secret -submit -jobname JOB_A -jobid 19561301 -start -wait

-cft_target_host pwdf0196 -cft_cmd_prefix "cmd /C" -cmdid 0000000001 -restart no

-cft_secure_cft no

Note: The Client Fault Tolerance Command ID Prefix is set up for a Windows host. In many user environments, the Client Fault Tolerance Command ID Prefix parameter can be specified in the configuration file and will never need to be specified on the command line. The same may be true for the Client Fault Tolerance Target Host parameter.

Secure CFT Mode

usap -user bob -pwd secret -submit -jobname JOB_A -jobid 19561301 -start -wait

-cft_secure_cft yes -cft_abap BTCTEST -cmdid 0000000001 -restart no

Note: In secure CFT mode, the **`cft_secure_cft`** and **`cft_abap`** parameters would most likely be specified in the USAP configuration file.

Restart of a Command ID Job

In the following example, USAP is requested to restart command ID job 0000000001. USAP will scan the SAP system for a matching command ID job.

If a matching command ID job is found, USAP reconnects to that job and satisfies the command line requirements. In this case, that means that if the job has not yet been started, it will be started, USAP will wait for the job to complete (if it hasn't already), and the output will be returned.

If no matching command ID job is found, USAP will terminate with an error code. Appropriate informational messages will be printed to standard error.

Pre-XBP 2.0 CFT Mode

usap -user bob -pwd secret -submit -jobname JOB_A -jobid 19561301 -start -wait

-cmdid 0000000001 -restart yes -cft_secure_cft no

Secure CFT Mode

usap -user bob -pwd secret -submit -jobname JOB_A -jobid 19561301 -start -wait

-cmdid 0000000001 -restart yes -cft_secure_cft yes

Run a Command ID Job Using Restart AUTO

In the following example, USAP will first scan the SAP system to determine if a matching command ID job exists. If no matching command ID job is found, USAP considers this to be the initial instance of this command ID job and defines the new command ID job to the SAP system. If a matching command ID job is found, USAP reconnects with the existing SAP job.

After determining if the command ID job is initial or a restart, USAP satisfies the command line requirements.

Pre-XBP 2.0 CFT Mode

usap -user bob -pwd secret -submit -jobname JOB_A -jobid 19561301 -start -wait

-cft_target_host pwdf0196 -cft_cmd_prefix "cmd /C" -cmdid 0000000001 -restart auto

-cft_secure_cft no

Secure CFT Mode

usap -user bob -pwd secret -submit -jobname JOB_A -jobid 19561301 -start -wait

-cmdid 0000000001 -restart auto -cft_secure_cft yes

2.6 z/OS CANCEL Command Support

Universal Connector provides Network Fault Tolerance and Client Fault Tolerance. These features provide users with the ability to execute jobs that will continue to run when the network is down and when USAP is terminated.

However, there are scenarios in which the user may want to cancel an executing job that supports client and / or network fault tolerance and have both the USAP and SAP processes terminate immediately. Because of the separation of work between USAP and SAP, when the USAP client is terminated, the SAP job continues to execute.

In particular, z/OS supports a CANCEL command that will terminate a job executing on the z/OS operating system. When a Universal Connector job is cancelled via the z/OS CANCEL command, the job terminates with either of these exit codes:

- S122, if job is cancelled with a dump.
- S222, if job is cancelled without a dump.

Part of the responsibility of a Universal Broker executing on a particular host is to monitor the status of all locally running manager processes on that machine. So, when instructed, that Universal Broker could invoke a new instance of Universal Connector and issue a CANCEL command to terminate the associated SAP job.

2.6.1 Exit Codes

Through the use of the SERVER_STOP_CONDITIONS configuration option, the Universal Connector process notifies the locally running Universal Broker of the exit codes that should cause it to terminate the running SAP job. With this option, you can specify a list of exit codes that should trigger the locally running Universal Broker to invoke a Universal Connector process to terminate the SAP job.

SERVER_STOP_CONDITIONS can specify a single exit code or a comma-separated list of exit codes. These stop conditions are passed from the manager to the locally running Universal Broker, which stores this and other component-specific data about the executing manager component. When this executing Universal Connector process is cancelled or stopped, the locally running Universal Broker detects the ending of the manager process and retrieves its process completion information, which includes the exit code of the manager.

The Universal Broker then compares this exit code with the list of exit codes provided by SERVER_STOP_CONDITIONS. If a match is found, the Universal Broker will invoke a new instance of the Universal Connector to execute a CANCEL command to terminate the running SAP job.

2.7 Mass Activities Support

Universal Connector supports the submission, starting, and monitoring of mass activities on the SAP system.To work with mass activities on the SAP system, Universal Connector utilizes the following SAP ABAP programs:

- **FKJO_SCHEDULE**
- **RFKK_MA_SCHEDULER**
- **RFKK_MASS_ACT_PARAMETER**

The basic process flow in working with mass activities is:

1. Create a template parameter record for the mass activity.
2. Copy the template parameter record and assign a Date ID and Run ID.
3. Schedule and start the mass activity.
4. Monitor the mass activity to completion.

The original template parameter records must be created on the SAP system using the dialogs for the given mass activity type. However, after a set of template parameter records have been created, Universal Connector can use the ABAP programs mentioned above to initiate and control the characteristics of mass activity work.

2.7.1 Initiating Mass Activities

Mass activities are initiated from Universal Connector by submitting and starting ABAP program **FKJO_SCHEDULE** or **RFKK_MA_SCHEDULER**. This can be accomplished by following the same procedure that would be used to submit and start any other ABAP program with Universal Connector. For more information on submitting and starting jobs with Universal Connector, see the SUBMIT, START, and RUN commands in Chapter 3 Universal Connector for z/OS and/or Chapter 4 Universal Connector for UNIX.

Both **FKJO_SCHEDULE** and **RFKK_MA_SCHEDULER** can be used to initiate mass activities. Each program has a different approach (and different requirements) for preparing a mass activity on the SAP system. The decision of which one to use must be made by understanding the capabilities and requirements of each program and matching those to the requirements of the situation. A discussion of the details of **FKJO_SCHEDULE** and **RFKK_MA_SCHEDULER** is beyond the scope of this document. For more information, please refer to the SAP documentation for these two programs.

The behavior of both **FKJO_SCHEDULE** and **RFKK_MA_SCHEDULER** are controlled by a set of parameters called a "variant." A variant is a named set of parameters that apply to a specific ABAP program. Variants reside on the SAP system. To achieve the desired results on a mass activity run, it may be necessary to modify the values of the variant used by the initiator program. In this case, initiating a mass activity becomes a two-step process. First, Universal Connector is used to create or modify an existing variant on the SAP system. Second, Universal Connector is used to submit and start the initiator program that uses the variant. For additional information on working with variants, refer to the SUBMIT VARIANT and MODIFY VARIANT commands in Chapter 3 Universal Connector for z/OS and/or Chapter 4 Universal Connector for UNIX.

2.7.2 Monitoring Mass Activities

Regardless of which program is used to initiate a mass activity, Universal Connector follows the same procedure for monitoring the process to completion. The MASS_ACTIVITY_WAIT command is used to instruct Universal Connector that it should perform this monitoring function (refer to the MASS_ACTIVITY_WAIT option in Chapter 3 Universal Connector for z/OS and/or Chapter 4 Universal Connector for UNIX).

Specifying the MASS_ACTIVITY_WAIT option will cause Universal Connector to monitor the status of the submitted/started job. In addition, as the jobs that make up the mass activity are created on the SAP system, Universal Connector detects them as child jobs of the initiator job and will begin to monitor their status as well. Universal Connector will continue to monitor the status of parent and child jobs until all jobs have completed.

Upon detecting the completion of a job, Universal Connector will optionally return the following information:

- Job log: see RETURN_JOB_LOG option.
- Application log (if one exists): see RETURN_APPLICATION_LOG option.
- Application return codes (if they were set): see RETURN_APPLICATION_RC option.
- Spooled output created by the job: see RETURN_SPOOL_LIST option.

In addition, Universal Connector will record the application return codes (if they are set) and merge them into its exit code mapping process that takes place upon program completion. Universal Connector will exit with the highest value used in the exit code processing.

2.7.3 Working with Parameter Records

With each mass activity run, there may be the need for parameter set adjustment. In some cases, the ABAP program used to initiate the mass activity can perform the necessary parameter adjustments. When more detailed parameter adjustments are required, the ABAP program **RFKK_MASS_ACT_PARAMETER** can be used. In this case, Universal Connector can be used to run **RFKK_MASS_ACT_PARAMETER** by following the same procedures that would be used to run any other ABAP program on the SAP system. For more information, refer to the SUBMIT, START, RUN, and WAIT commands in Chapter 3 Universal Connector for z/OS and/or Chapter 4 Universal Connector for UNIX.

The information that controls how **RFKK_MASS_ACT_PARAMETER** will adjust the mass activity parameter set is contained in a variant that resides on the SAP system. In many cases, it may be necessary to create or modify the contents of a variant with information that pertains to a specific mass activity. In this case, Universal Connector can be used to create or modify the variants as needed. For additional information on working with variants, refer to the SUBMIT VARIANT and MODIFY VARIANT commands in Chapter 3 Universal Connector for z/OS and/or Chapter 4 Universal Connector for UNIX.

2.8 CM Interface

USAP's CM Interface (USAP-CM) serves the purpose of providing functionality that is not possible using the certified SAP XBP interface alone.

USAP-CM offers an external interface to MHP's Communication Management for SAP (CM). Specifically, this provides the ability to work with CM Communications and CM Report jobs.

Communication Management is a licensed product from MHP. Please contact your MHP representative for information on the required CM prerequisites for the USAP CM Interface.

2.8.1 CM Communications

CM communications are capable of controlling complex job nets including communications between SAP systems and non-SAP systems. With USAP-CM, you can start and monitor CM communications that have been defined within the SAP system and return spooled output, joblog, and communication log upon completion. USAP-CM also has the ability to restart terminated CM communications. Restarts can be performed from the point of failure or, skipping the failed component.

CM communications are defined in the SAP system using the MHP Communication Management application. The structures that determine the processing steps that are carried out by a CM communication cannot be created or altered by USAP-CM. However, many attributes that affect the characteristics of a communication can be set using USAP-CM. These characteristics include an identification string (CM OPC Token), output control (spoollists and logs), printing and archiving options, and the start mode of a CM communication.

The user definable attributes that are used to control a CM Communication are specified in job definition files. This document refers to the collection of attributes as a CM Communication Job Definition.

If changes are required to CM communication attributes, the CM communication is started using the SUBMIT command referencing an appropriate job definition file

If no changes are required to the CM communication attributes defined in the SAP system, a CM communication can be started with the START command and a job definition file would not be required. In this case, a CM Communication Id and CM OPC Token could be specified on the USAP command line.

USAP Command Groups

The following USAP command groups can be used with CM report jobs: SUBMIT, START, RUN, and WAIT.

- A Communication ID is used to reference CM Communications stored in an SAP system. The Communication ID can be specified in a job definition file (keyword CM_COMMUNICATION_ID) and on the command line (parameter **`-cmid`**).
- A user defined text string is used to prefix SAP jobs created to process a CM Communication. This text string can be specified in the job definition (keyword CM_OPC_TOKEN) and on the command line (parameter **`-cmopctoken`**).
- The CM Communication start mode can be specified in the job definition file (keyword CM_MODE) and on the command line (parameter **`-cmmode`**).

If parameters **`-cmid`**, **`-cmopctoken`**, or **`-cmmode`** are specified on the command line, their values override any that may have been defined in the job definition file.

Exit Codes For CM Communications

The exit code of USAP depends on whether or not the **`-wait`** option is used to wait for CM Communication completion. If waiting for CM Communication completion, the exit code represents the CM return code for the CM communication's status upon completion.

Table 2.2, below, identifies the exit codes that indicate the CM communication status.

Exit Code	Job Status
0	CM communication completed successfully.
1 – 11	Warnings and error messages generated by the processed communication components.
12	Error occurred attempting to process a communication component.
> 200	Error in USAP processing (see Table 2.3).

Table 2.2 CM Communication Exit Codes

If USAP is not waiting for CM communication completion, the exit code indicates the success of the requested actions.

Table 2.3, below, identifies the USAP exit codes.

Exit Code	Description
0	Successfully completed all requested actions.
201	Error occurred processing the requested actions. Messages are printed providing details about the error.
210	Error with product configuration options or command line options.
211	Error occurred in the initialization phase of message processing. It is possible that the error prohibited messages from printing.

Table 2.3 USAP Exit Codes

Sample Command Lines For Working With CM Communications

usap -user *userid* -pwd *password* -submit *filename* -wait

This example would submit communication attributes defined in *filename*, start, and wait for the CM communication to complete. The value for the CM OPC token is provided in the job definition file *filename*.

usap -user *userid* -pwd *password* -submit *filename* -cmopctoken *12345678* -wait

This example would perform the same actions as the previous command line; submit, start, and wait for the CM communication defined in job definition file *filename*. The value specified on the command line for **`-cmopctoken`** will override any value that may have been supplied in the job definition file.

usap -user *userid* -pwd *password* -run *filename* -cmopctoken *12345678*

The above example would perform the same actions as the previous command line; submit, start, and wait for the CM communication defined in job definition file **`filename`**. The value specified on the command line for **`-cmopctoken`** will override any value that may have been supplied in the job definition file. This command line takes advantage of the RUN shortcut command.

usap -user *userid* -pwd *password* -run -cmopctoken *12345678* -cmid *COM1* -cmmode *N*

The above example uses the RUN shortcut command to start, and wait for the CM communication identified by parameter **`-cmid`**. No attributes of the communication are changed and, therefore, no job definition file was needed.

2.8.2 CM Report Jobs

CM Report jobs are SAP jobs that run a single ABAP report. USAP-CM submits and starts CM report jobs on the SAP system using CM function modules within SAP. CM report jobs offer several advantages over standard SAP jobs defined using XBP. CM report jobs allow control of job attributes that are not available through XBP. These attributes include job class, variable variant parameters, and certain print and archiving parameters.

CM report jobs have some unique behaviors and characteristics that differ from standard XBP jobs defined and started with USAP. These include the following:

1. CM jobs require a unique token. This unique token can be specified in the CM report job definition file with keyword CM_OPC_TOKEN and on the command line with parameter **`-cmopctoken`**. If the token is defined in both the job definition file and the command line, the command line value will override the value specified in the job definition file.
2. When a CM report job is defined to an SAP system, a job will be created with a jobname that has been generated with the following pattern: CM_OPC_TOKEN-CM_REPORT. That is to say that the job name will be the ABAP report name that is being run, prefixed with the value supplied for the CM_OPC_TOKEN parameter, and the two values being separated by a dash (-).
3. When a CM report job is submitted to the SAP system, a new job is created and started. It is, therefore, not possible to schedule a CM report job without starting it.
4. If the WAIT command is used when submitting a CM report job, the CM_OPC_TOKEN/CM_REPORT combination must produce a unique jobname in the SAP system. Before performing the WAIT operation, USAP will query the SAP system for the newly generated job name. If multiple jobs are found with the same jobname, USAP reports this condition and ends with an error code.

The following USAP command groups can be used with CM report jobs: SUBMIT, RUN, and WAIT.

CM report jobs are defined using a USAP-CM CM Report job definition file. A CM Report job definition file contains statements that describe a CM Report job. These statements are used by the SUBMIT and RUN commands to define and start a CM Report job on an SAP system.

Exit Codes For CM Report jobs

The exit code processing for CM Report jobs is the same as for Standard USAP jobs. For details, see Sections 3.5 Exit Codes (z/OS) and 4.5 Exit Codes (UNIX).

Sample Command Lines For Working With CM Report Jobs

usap -user *userid* -pwd *password* -submit *filename* -cmopctoken *12345678* -wait

This example would submit, start, and wait for the CM Report job defined in job definition file *filename*. The value specified on the command line for **`-cmopctoken`** will override any value that may have been supplied in the job definition file.

usap -user *userid* -pwd *password* -run *filename* -cmopctoken *12345678*

This example would perform the same actions as the previous command line; submit, start, and wait for the CM report job defined in job definition file *filename*. The value specified on the command line for **`-cmopctoken`** will override any value that may have been supplied in the job definition file. This command line takes advantage of the RUN shortcut command.

usap -user *userid* -pwd *password* -run *filename*

This example uses the RUN shortcut command to submit, start, and wait for the CM report job defined in job definition file *filename*. The **`cmopctoken`** value is specified in the job definition file.

2.8.3 Auto CM Option

New keywords have been added to the standard USAP job definition. These keywords are used to specify job parameters that are not supported by XBP 1.0. The use of these keywords requires a license for the CM Interface. See Section 5.4 Keywords for ABAP Step Statement for the keywords available to standard USAP job definitions.

If one or more of these new (non-XBP) keywords are specified in a job definition file, USAP can automatically use the CM interface to submit and start the job. The resulting job will be generated using the same CM function modules used for the CM Report jobs described in the previous section. As a result, the job will have the same unique characteristics of CM Report jobs described in the previous section.

The Auto CM option is used to turn on and off the automatic use of the CM interface for standard USAP jobs. The Auto CM option can be specified in the USAP configuration file (keyword **`auto_cm`**) or on the command line (option **`-autocm`**). Both parameters accept a value of yes or no.

2.9 Batch Input Monitoring

USAP supports the monitoring of batch input session processing. This support is currently limited to SAP 4.6C and above. To perform batch input monitoring, USAP utilizes the functionality of SAP's ABAP program **RSBDCSUB**.

RSBDCSUB selects batch input sessions for processing based on the criteria specified in its variant. The batch input sessions selected to be processed by **RSBDCSUB** are transferred to the SAP system's background processing. **RSBDCSUB** completes independent of the session processing jobs it starts. The spoollist produced by **RSBDCSUB** contains the information required to identify the session processing jobs created, and relate them to their respective batch input sessions. This information consists of a job name (same as session name), job id, and queue id. The job name, job id combination uniquely identifies the session processing job and the queue id uniquely identifies the queue that contains the batch input session data and status.

2.9.1 Batch Input Monitoring Process

A basic overview of the USAP batch input monitoring process follows:

1. USAP starts a single step job that executes ABAP program **RSBDCSUB**, or USAP connects to a previously started single step job executing ABAP program **RSBDCSUB**.
2. USAP waits for the **RSBDCSUB** job to complete.
3. If the **RSBDCSUB** job terminates, USAP exits with the USAP 'Terminated' job status code. Otherwise, USAP retrieves the spoollist generated by **RSBDCSUB** and extracts the session processing information. This information consists of the session processing jobs that were kicked off by **RSBDCSUB**, and the corresponding queues that contain the sessions.
4. USAP begins to monitor all session processing jobs that were kicked off by **RSBDCSUB**. When USAP detects that a session processing job has completed, it retrieves the state of the corresponding queue and converts the queue state to a USAP queue state exit code. USAP continues this monitoring process until all session processing jobs have completed.
5. When all session processing jobs have completed, USAP exits with the highest queue state exit code retrieved from all sessions that were processed by **RSBDCSUB**.

2.9.2 Batch Input Monitoring Requirements

SAP System

USAP only supports batch input monitoring on SAP 4.6 systems. This restriction is based on the ABAP program RSBDCSUB. RSBDCSUB kicks off session processing jobs and completes independent of the session processing jobs. Only the SAP 4.6 version of RSBDCSUB produces a spoollist that contains all the information needed to monitor the session processing jobs and the states of the sessions they process. This information consists of the job name and job id of the session processing jobs that get kicked off, and the queue id of the session that is being processed.

SAP Batch Input Sessions

All batch input sessions that will be monitored by USAP must have the **`keep session`** flag checked. This is required because the queue that contains the batch input session must exist in the SAP system after the session processing job completes in order for USAP to retrieve the state of the queue.

USAP

To perform batch input monitoring with USAP, a single step SAP job must be started that executes ABAP program RSBDCSUB. USAP can start the job or can connect to a job that was previously started.

USAP uses the spoollist generated by RSBDCSUB to extract session processing information. The format of this report depends on the language of the job step. There are three USAP parameters that must be set up for the language being used. See Sections 3.3.15 BDCWAIT Command (z/OS) and 4.3.15 BDCWAIT Command (UNIX) for details. By default, these parameters are set up to work with the English language.

The print parameters for the job step executing RSBDCSUB must specify enough columns to allow the full width of the report to be generated without truncation. A value of 132 is sufficient. In addition, the number of lines per page must allow the entire report to be generated on a single page. This is due to limitations in the RSBDCSUB report generation capability.

The USAP command line parameter **`-bdcwait`** is used to initiate the batch input monitoring process. For details on this parameter, see Sections 3.3.15 BDCWAIT Command (z/OS) and 4.3.15 BDCWAIT Command (UNIX).

2.9.3 Batch Input Monitoring Parameters

The set of USAP configuration parameters that are specific to the batch input monitoring support are:

- BDC Wait
- BDC Job Name Pattern
- BDC Job ID Pattern
- BDC Queue ID Pattern
- Queue "to be created" exit code mapping
- Queue "unprocessed" exit code mapping
- Queue "in background" exit code mapping
- Queue "finished" exit code mapping
- Queue "error" exit code mapping

See Sections 3.3.15 BDCWAIT Command (z/OS) and 4.3.15 BDCWAIT Command (UNIX) for details concerning the use of these parameters.

2.10 Message and Audit Facilities

All Universal Products have the same message facilities. Messages - in this context - are text messages written to a console, file, or system log that:

1. Document the actions taken by a program.
2. Inform users of error conditions encountered by a program.

This section describes the message and audit facilities that are common to all Universal Products. (See the individual Universal Product documentation for additional details.)

2.10.1 Message Types

There are six types (or severity levels) of Universal Products messages. (The severity level is based on the type of information provided by those messages.)

1. Audit messages document the configuration options used by the program's execution and resource allocation details. They provide complete description of the program execution for auditing and problem resolution.
2. Informational messages document the actions being taken by a program. They help determine the current stage of processing for a program. Informational messages also document statistics about data processed.
3. Warning messages document unexpected behavior that may cause or indicate a problem.
4. Error messages document program errors. They provide diagnostic data to help identify the cause of the problem.
5. Diagnostic messages document diagnostic information for problem resolution.
6. Alert messages document a notification that a communications issue, which does not disrupt the program or require action, has occurred.

The MESSAGE_LEVEL configuration option in each Universal Product component lets you specify which messages are written (see Section 2.10.3 Message Levels).

2.10.2 Message ID

Each message is prefixed with a message ID that identifies the message.

The message ID format is **UNVnnnnl**, where:

- **nnnn** is the message number.
- **l** is the message severity level:
  - **A** (Audit)
  - **I** (Informational)
  - **W** (Warning)
  - **E** (Error)
  - **T** (alerT)
  - **D** (Diagnostic)

Note: The Universal Products 4.1.0 Messages and Codes document identifies all messages numerically, by product, using the **nnnn** message number.

2.10.3 Message Levels

Each Universal Product includes a MESSAGE_LEVEL configuration option that lets you select which levels (that is, severity levels) of messages are to be written.

- *Audit* specifies that all audit, informational, warning, and error messages are to be written.
- *Informational* specifies that all informational, warning, and error messages are to be written.
- *Warning* specifies that all warning and error messages are to be written.
- *Error* specifies that all error messages are to be written.
- *Trace* specifies that a trace file is created, to which data used for program analysis will be written. The trace file name and location are Universal Product dependent (see the appropriate Universal Product documentation for details).

  (Trace should be used only at the request of Stonebranch, Inc. Customer Support.)

Note: Diagnostic and Alert messages always are written, regardless of the level selected in the MESSAGE_LEVEL option.

2.10.4 Message Destinations

The location to which messages are written is the message destination.

Some Universal Products have a MESSAGE_DESTINATION configuration option that specifies the message destination. If a program is used only from the command line or batch job, it may have only one message destination, such as standard error.

Valid message destination values depend on the host operating system.

z/OS Message Destinations

Universal Products on z/OS run as batch jobs or started tasks. Batch jobs do not provide the MESSAGE_DESTINATION option. All messages are written to the SYSOUT ddname.

Started task message destinations are listed in the table below.

Destination	Description
LOGFILE	Messages are written to ddname UNVLOG. All messages written to log files include a date and time stamp and the program's USS process ID.
SYSTEM	Messages are written to the console log as WTO messages.

UNIX Message Destinations

Message destinations are listed in the table below.

Destination	Description
STDERR	Messages are written to standard error. This destination is most useful for console commands.
LOGFILE	Messages are written to a log file. Not all programs provide this destination. The recommended directory for log files is **`/var/opt/universal/log`**. This can be changed with the LOG_DIRECTORY option. All messages written to log files include a date and time stamp and the program's process ID.
SYSTEM	Messages are written to the syslog daemon. Not all programs provide this destination. Universal programs that execute as daemons write to the syslog's daemon facility. All messages include the programs process ID. If an error occurs writing to the syslog, the message is written to the system console.

Windows Message Destinations

Message destinations are listed in the table below.

Destination	Description
STDERR	Messages are written to standard error. This destination is most useful for console commands.
LOGFILE	Messages are written to a log file. Not all programs provide this destination. Log files are written to product specific log directories, which can be modified with the LOG_DIRECTORY option. All messages written to log files include a date and time stamp and the program's process ID.
SYSTEM	Messages are written to the Windows Application Event Log.

OS/400 Message Destinations

Message destinations are listed in the table below.

Destination	Description
STDERR	Messages are written to standard error. A batch job's standard error file is allocated to the print file QPRINT.
LOGFILE	Messages are written to the job's job log.
SYSTEM	Messages are written to the system operator message queue QSYSOPR.

HP NonStop Message Destinations

Message destinations are listed in the table below.

Destination	Description
STDERR	Messages are written to standard error.
LOGFILE	Messages are written to a log file. Not all programs provide this destination. Log files are written the **$SYSTEM.UNVLOG** subvolume. All messages written to log files include a date and time stamp and the program's process ID.

Chapter 3
Universal Connector for z/OS

3.1 Overview

This chapter documents Universal Connector (USAP) at a detailed level, specific to the z/OS operating system.

It provides sections of the following information:

- Usage
- Commands
- Configuration Options
- Exit Codes

3.2 Usage

Universal Connector for z/OS executes as a batch job.

Each batch job contains:

1. JCL interface to the command line.
2. Configuration options associated with the specified command.
3. Configuration options (required and optional) not associated with any specific command.

Universal Connector performs an operation specified by the command. The configuration options describe the actions to take for that operation.

This section describes the JCL and command line syntax of Universal Connector for z/OS.

3.2.1 JCL Procedure

Figure 3.1, below, illustrates the Universal Connector for z/OS JCL procedure (**USPPRC**, located in the **SUSPSAMP** library) that is provided to simplify the execution JCL and future maintenance.

```
//USPPRC   PROC UPARM=,                -- USAP options
//             SAPRFC=USPRFC00,        -- SAP RFC member
//             USAPPRE=#SHLQ.UNV,
//             USAPPRD=#PHLQ.UNV
//*
//PS1      EXEC PGM=USAP,PARM='ENVAR(TZ=EST5EDT)/&UPARM'
//STEPLIB  DD  DISP=SHR,DSN=&USAPPRE..SUNVLOAD
//*
//UNVNLS   DD  DISP=SHR,DSN=&USAPPRE..SUNVNLS
//UNVRFC   DD  DISP=SHR,DSN=&USAPPRD..UNVCONF(&SAPRFC)
//UNVTRACE DD  SYSOUT=*
//*
//SYSPRINT DD  SYSOUT=*
//SYSOUT   DD  SYSOUT=*
//CEEDUMP  DD  SYSOUT=*
```

Figure 3.1 Universal Connector for z/OS – JCL Procedure

The parameter **UPARM** specifies EXEC PARM keyword values. The parameter **CONFIG** specifies the configuration member. The parameter **SAPRFC** specifies the SAP RFC configuration member. The parameter **USAPPRE** specifies the data set name prefix of USAP installation data sets.

3.2.2 DD Statements used in JCL Procedure

Table 3.1, below, describes the DD statements used in the Universal Connector for z/OS JCL illustrated in Figure 3.1.

ddname	DCB Attributes *	Mode	Description
STEPLIB	DSORG=PO, RECFM=U	input	Load library containing the program being executed.
UNVNLS	DSORG=PO, RECFM=(F, FB, V, VB)	input	USAP national language support library. Contains message catalogs.
UNVRFC	DSORG=PS, RECFM=(F, FB, V, VB)	input	SAP Remote Function Call (RFC) configuration member.
UNVTRACE	DSORG=PS, RECFM=(F, FB, V, VB)	Output	USAP trace output.
SYSPRINT	DSORG=PS, RECFM=(F, FB, V, VB)	output	Standard output file for the USAP program.
SYSOUT	DSORG=PS, RECFM=(F, FB, V, VB)	output	Standard error file for the USAP program.
* The C runtime library determines the default DCB attributes. Refer to the IBM manual *OS/390 C/C++ Programming Guide* for details on default DCB attributes for stream I/O			

Table 3.1 Universal Connector for z/OS – DD Statements in JCL Procedure

3.2.3 JCL

Figure 3.2, below, illustrates the Universal Connector for z/OS JCL using the **USPPRC** procedure illustrated in Figure 3.1.

```
//jobname  JOB CLASS=A,MSGCLASS=X
//STEP1    EXEC USPPRC
//SYSIN    DD  *
 -dest BIN_45 -client 850 -userid user -pwd password
 -run
 -jobname SAMPLE1
 -jobid   13203001
 . . .
/*
```

Figure 3.2 Universal Connector for z/OS – JCL

Job step STEP1 executes the procedure **USAPPRC**.

The command options are specified on the SYSIN DD.

3.2.4 Command Line Syntax

Figure 3.3, below, illustrates the command line syntax of Universal Connector for z/OS.

```
usap  {RUN | SUBMIT | MODIFY | START | WAIT | ABORT | DISPLAY | GENERATE |
      PURGE | SYNTAX | RAISE EVENT} HOST USER [CFT] [EVENT] [INFORMATIONAL]
      [LOCAL] [MESSAGE] [RFC] [COMMAND FILE]
```

Figure 3.3 Universal Connector for z/OS - Command Line Syntax

Names enclosed in {BRACES} identify command groups. For each execution, a single command is specified from one of these groups. One or more configuration options associated with each command also can be used to specify additional information / actions for the execution.

Names encoded in [BRACKETS] identify categories of configuration options that are not associated with specific commands and from which options are not required.

Names not enclosed in {BRACES} or [BRACKETS] identify categories of configuration options that are not associated with specific commands but from which one or more options are required.

See Section 3.3 Commands for detailed information on the commands, and their associated configuration options, in each command group.

See Section 3.4 Configuration Options for detailed information on configuration options not associated with one or more commands.

Example

The following is an example of a command line syntax executing Universal Connector:

```
usap -sub file.usp -immediate -client 987 -dest -userid 123 -pwd ABC
-lang english -level info
```

3.3 Commands

This section identifies all of the Universal Connector for z/OS commands.

Each command has configuration options associated with it that can be used to specify additional information / actions for an execution of that command.

(For detailed information on configuration options not associated with one or more specific commands, see Section 3.4 Configuration Options.)

3.3.1 Command Groups

Universal Connector groups commands into areas of common functionality, as shown in Table 3.2, below. Each row identifies a command group, the commands in that group, and the type of operation performed by those commands.

The name of each command is a link to the following information in this section:

- Command description — Description of the operation(s) performed by the command.
- Command line syntax — Syntax of the command and its options on the command line.
- Command argument — Command line expression (short and/or long form) and description of the command argument.
- Command options — Description of the configuration options associated with the option and a link to detailed information about those options in the Universal Connector 4.1.0 Reference Guide.

Command Groups	Description
RUN • RUN JOB Command • RUN FS JOB NETWORK Command	Performs the following actions: 1. Defines an SAP job. 2. Starts the job. 3. Waits for the job to complete. 4. Writes the joblog and spoollists of the job. 5. Purges the job from the SAP system.
SUBMIT • SUBMIT FS JOBNET Command • SUBMIT JOB Command • SUBMIT INTERCEPT CRITERIA TABLE Command • SUBMIT VARIANT Command	Defines a job to the SAP system.
MODIFY • MODIFY JOB Command • MODIFY VARIANT Command	Modifies a job in an SAP job.
START • START FS JOBNET Command • START JOB Command	Starts a defined SAP job.

Command Groups	Description
WAIT • WAIT for JOB Command • MASS ACTIVITY WAIT Command • WAIT for FS JOB NETWORK Command • BDCWAIT Command	Allows USAP to reconnect to a started job and monitor it through completion.
ABORT • ABORT Command	Cancels a running SAP job.
DISPLAY • DISPLAY COMMANDS Command • DISPLAY INTERCEPTED_JOBS Command • DISPLAY INTERCEPT_TABLE Command • DISPLAY JOBDEF Command • DISPLAY JOBLOG Command • DISPLAY OUTPUT_DEVICES Command • DISPLAY PRINT_FORMATS Command • DISPLAY QSTATE Command • DISPLAY REPORTS Command • DISPLAY SPOOLLIST Command • DISPLAY STATUS Command • DISPLAY VARIANT Command • DISPLAY VARIANTS Command • DISPLAY SELECT Command • DISPLAY SYSLOG Command	Displays various SAP job data.
GENERATE • GENERATE JOB DEFINITION FILE Command • GENERATE VARIANT DEFINITION FILE Command	Generates USAP job or variant definitions based on model SAP jobs or variants.
PURGE • PURGE JOB Command • PURGE FS JOB NETWORK Command	Deletes SAP jobs.
RAISE EVENT Command	Raises the specified SAP background processing event.
SYNTAX Command	Checks the syntax of a job definition file.

Table 3.2 USAP for z/OS - Command Groups

3.3.2 RUN JOB Command

The RUN JOB command performs the following actions:

1. Defines a new SAP, job based on either a job definition specification or an existing SAP job definition.
2. Starts the defined job.
3. Waits for the job to complete.
4. Prints the job's joblog to standard error and the spoollists to standard output.
5. Purges the job from the SAP system.

The exit code of USAP will indicate the completion status of the SAP job.

See Section 3.5.1 WAIT for JOB Exit Codes for a complete list of job status exit codes.

RUN JOB Command Line Syntax

Figure 3.4, below, illustrates the command line syntax – using the command line, long form of the configuration options – of the RUN JOB command.

```
-run {ddname | -jobname jobname –jobid jobid}
    [-target_jobname jobname]
    [-poll seconds]
    [-job_stat_check_interval seconds]
    [-targetserver server]
    [-target_variant job step,variant name;job step,variant name;...]
    [-immediate]
    [-cmopctoken token]
    [-cmdid id]
    [-cmmode {N|R|S|T}]
    [-activeec exitcode]
    [-readyec exitcode]
    [-scheduledec exitcode]
    [-releasedec exitcode]
    [-terminatedec exitcode]
    [-finishedec exitcode]
    [-max_log_size size]
    [-max_spool_size size]
    [-server_stop_conditions codes]
    [-spool_codepage codepage]
    [-bdcwait
         [-bdcjobnameptrn pattern]
         [-bdcjobidptrn pattern]
         [-bdcqidptrn pattern]
         [-qtobecreatedec exitcode]
         [-qunprocessedec exitcode]
         [-qinbackgroundec exitcode]
         [-qfinishedec exitcode]
         [-qerrorec exitcode]
    ]
```

Figure 3.4 RUN JOB Command Line Syntax

RUN JOB Command Argument

The RUN JOB command can be expressed as either:

- -R Short form
- -run Long form

The RUN JOB command argument, *ddname*, specifies the name of the file that contains the job definition.

See Chapter 5 Job Definition Files for additional information on the job definition file.

RUN JOB Command Options

Option Name	Description
BATCH_MONITOR	Causes USAP to perform batch input monitoring for the started job.
CM_ID	Name of the CM communication to be started.
CM_MODE	Starting mode of a CM Communication.
CM_TOKEN	Unique character string used to prefix SAP jobs.
EXIT_JOB_ACTIVE	USAP exit code for the SAP job active status.
EXIT_JOB_FINISHED	USAP exit code for the SAP job finished status.
EXIT_JOB_READY	USAP exit code for the SAP job ready status.
EXIT_JOB_RELEASED	USAP exit code for the SAP job released status.
EXIT_JOB_SCHEDULED	USAP exit code for the SAP job scheduled status.
EXIT_JOB_TERMINATED	USAP exit code for the SAP job terminated status.
EXIT_QUEUE_BACKGROUND	USAP exit code for the SAP queue state **S** (in background).
EXIT_QUEUE_CREATED	USAP exit code for the SAP queue state **C** (to be created).
EXIT_QUEUE_ERROR	USAP exit code for the SAP queue state **E** (error).
EXIT_QUEUE_FINISHED	USAP exit code for the SAP queue state **F** (finished).
EXIT_QUEUE_UNPROCESSED	USAP exit code for the SAP queue state [ ] (unprocessed).
IMMEDIATE_JOB	Causes the job to be started immediately.
JOB_ID	Job ID of an existing SAP job to use as a model for the new job definition.
JOB_ID_PATTERN	Locates the header record and determines the offset of the job ID in the RSBDCSUB batch input processing report.
JOB_NAME	Existing SAP job name to use as a model for the new job definition.
JOB_NAME_PATTERN	Locates the header record and determines the offset of the job name in the RSBDCSUB batch input processing report.
MAX_JOB_LOG_SIZE	Maximum size for job logs.
MAX_SPOOL_LIST_SIZE	Maximum size for spool lists.
QUEUE_ID_PATTERN	Locates the header record and determines the offset of the queue ID in the RSBDCSUB batch input processing report.
SERVER_STOP_CONDITIONS	Exit code(s) of the executing Universal Connector process that should trigger the locally running Universal Broker to cancel the corresponding SAP job.
SPOOL_CODEPAGE	Codepage used for transferring spool lists from SAP system.
STATUS_CHECK_INTERVAL	Length of time that can elapse, without a change in job status, before a call will be made to synchronize the actual job status with the SAP stored status.
TARGET_JOB_NAME	Name to give the newly created job.
TARGET_SERVER	Server on which the job will run.
TARGET_VARIANT	One or more replacement variants for ABAP program job steps in an SAP job.
USAP_POLL	Length of time to wait between job status calls to the SAP system.

Table 3.3 RUN JOB Command Options

3.3.3 RUN FS JOB NETWORK Command

The RUN FS JOB NETWORK command performs the following actions:

1. Defines a new SAP FS job network based on a USAP FS Job Network definition file.
2. Starts the defined FS job network.
3. Waits for the started FS job network to complete.
4. Purges the FS job network from the SAP system.

The exit code of **usap** will indicate the completion status of the FS job network.

See Section 3.5.2 WAIT for FS JOB NETWORK Exit Codes for a complete list of job status exit codes.

RUN FS JOB NETWORK Command Line Syntax

Figure 3.5, below, illustrates the command line syntax – using the command line, long form of the configuration options – of the RUN FS JOB NETWORK command.

```
-run {ddname | -jnetid jobnetid -jnetprcid processid}
```

Figure 3.5 RUN FS JOB NETWORK Command Line Syntax

RUN FS JOB Command Argument

The RUN FS JOB command can be expressed as either:

- -R Short form
- -run Long form

The RUN FS JOB command argument, *ddname*, specifies the name of the file that contains the FS job network definition.

See Section 5.12 FS Job Network Definition File for additional information on the FS job network definition file.

RUN FS JOB NETWORK Command Options

Option Name	Description
JOB_NETWORK_ID	Network identifier for the pre-existing SAP FS job network being started.
JOB_PROCESS_ID	Process ID of an existing SAP FS job network process to start.

Table 3.4 RUN FS JOB NETWORK Command Options

3.3.4 SUBMIT JOB Command

The SUBMIT JOB command defines a new SAP job.

SUBMIT JOB Command Line Syntax

Figure 3.6, below, illustrates the command line syntax – using the command line, long form of the configuration options – of the SUBMIT JOB command.

```
-sub {ddname | -jobname jobname -jobid jobid}
    [-target_jobname jobname]
    [-start
        [-immediate]
        [-targetserver server]
        [-target_variant job step,variant name;job step,variant name;...]
        [-wait
            [-poll seconds]
            [-joblog {yes|no}]
            [-spoollist {yes|no}]
            [-purge]
            [-waitchild {yes|no}]
            [-max_child_depth depth]
            [-joblogchild {yes|no|error}]
            [-spoollistchild {yes|no}]
            [-purgechild {yes|no}]
        ]
    ]
```

Figure 3.6 SUBMIT JOB Command Line Syntax

SUBMIT JOB Command Argument

The SUBMIT JOB command can be expressed as either:

- -U Short form
- -sub Long form

The SUBMIT JOB command argument, *ddname*, specifies the name of the file that contains the job definition.

See Chapter 5 Job Definition Files for additional information on the job definition file.

SUBMIT JOB Command Options

Option Name	Description
IMMEDIATE_JOB	Causes the job to be started immediately.
JOB_ID	Job ID of an existing SAP job to use as a model for the new job definition.
JOB_LOG_CHILD	Controls the printing of job logs for child jobs.
JOB_NAME	Name of an existing SAP job to use as a model for the new job definition.
MAX_CHILD_DEPTH	Controls the maximum relationship depth that will be monitored by USAP.
PURGE_CHILD_JOBS	Controls the purging of child jobs.
PURGE_JOB	Purge job that has completed processing from SAP system.
RETURN_JOB_LOG	Specification for whether or not the job's joblog is returned.
RETURN_SPOOL_LIST	Specification for whether or not the spoollists of all job steps are returned.
SPOOL_LIST_CHILD	Controls the printing of spoollists for child jobs.
START_JOB	Starts the newly defined job.
TARGET_JOB_NAME	Name to give the newly created job.
TARGET_VARIANT	One or more replacement variants for ABAP program job steps in an SAP job.
WAIT	Wait for the SAP job to complete processing.
WAIT_FOR_CHILD_JOBS	Controls the monitoring of child jobs.

Table 3.5 SUBMIT JOB Command Options

3.3.5 SUBMIT VARIANT Command

The SUBMIT VARIANT command defines a new variant to an SAP system for a specified ABAP report.

Note: SUBMIT VARIANT requires XBP interface 2.0.

(See Section 2.5 Client Fault Tolerance for information on XBP interface 2.0.)

SUBMIT VARIANT Command Line Syntax

Figure 3.7, below, illustrates the command line syntax – using the command line, long form of the configuration options – of the SUBMIT VARIANT command.

```
-sub ddname
```

Figure 3.7 SUBMIT VARIANT Command Line Syntax

SUBMIT VARIANT Command Argument

The SUBMIT VARIANT command can be expressed as either:

- -U Short form
- -sub Long form

The SUBMIT VARIANT command argument, *ddname*, specifies the name of the file that contains the variant definition.

See Section 5.10 Variant Definition File for additional information on the variant definition file.

3.3.6 SUBMIT INTERCEPT CRITERIA TABLE Command

The SUBMIT INTERCEPT CRITERIA TABLE command appends or replaces the SAP intercept criteria table.

SUBMIT INTERCEPT CRITERIA TABLE Command Line Syntax

Figure 3.8, below, illustrates the command line syntax – using the command line, long form of the configuration options – of the SUBMIT INTERCEPT CRITERIA TABLE command.

```
-sub ddname
```

Figure 3.8 SUBMIT INTERCEPT CRITERIA TABLE Command Line Syntax

SUBMIT INTERCEPT CRITERIA TABLE Command Argument

The SUBMIT INTERCEPT CRITERIA TABLE command can be expressed as either:

- -U Short form
- -sub Long form

The SUBMIT INTERCEPT CRITERIA TABLE command argument, *ddname*, specifies the name of the file that contains the intercept criteria table definition.

See Section 5.11 Job Intercept Table Definition File for additional information on the variant definition file.

3.3.7 SUBMIT FS JOBNET Command

The SUBMIT FS JOBNET command defines a new FS jobnet to an SAP system.

SUBMIT FS JOBNET Command Line Syntax

Figure 3.9, below, illustrates the command line syntax – using the command line, long form of the configuration options – of the SUBMIT FS JOBNET command.

```
-sub {ddname | -jobname jobname -jobid jobid}
    [-start
        [-wait
             [-poll seconds]
             [-purge]
        ]
    ]
```

Figure 3.9 SUBMIT FS JOBNET Command Line Syntax

SUBMIT FS JOBNET Command Argument

The SUBMIT FS JOBNET command can be expressed as either:

- -U Short form
- -sub Long form

The SUBMIT FS JOBNET command argument, *ddname*, specifies the name of the file that contains the FS jobnet definition.

See Section 5.12 FS Job Network Definition File for additional information on the variant definition file.

SUBMIT FS JOBNET Command Options

Option Name	Description
PURGE_JOB	Purge job that has completed processing from SAP system.
START_JOB	Starts the newly defined job.
USAP_POLL	Length of time to wait between job status calls to the SAP system.
WAIT	Wait for the SAP job to complete processing.

Table 3.6 SUBMIT FS JOBNET Command Options

3.3.8 MODIFY JOB Command

The MODIFY JOB command is used to modify an SAP job that already exists in an SAP system. A USAP job definition file is used to specify the modifications.

Job definition files are used to define new SAP jobs and to modify existing SAP jobs. The same syntactical rules apply to the job definition file in both cases with the following exceptions when modifying jobs:

1. SAP job identifier must be specified in order to identify the existing job to modify. The job identifier is specified in the job definition file using the JOBCOUNT keyword of the Job Header statement or the **-jobid** option of the MODIFY command. If both are used, the **-jobid** option overrides the JOBCOUNT value.
2. ABAP Step and External Step job definition statements must specify the step number of the existing job step to modify. The step number is specified using the STEP_NUMBER keyword of the ABAP Step and External Step job definition statements.

The parameter values specified in job definition file replace existing values in the SAP job definition. If a parameter is not specified in the job definition file, no change is made to the corresponding value in the existing SAP job definition.

See Chapter 5 Job Definition Files for additional information on the job definition file.

MODIFY JOB Command Line Syntax

Figure 3.10, below, illustrates the command line syntax – using the command line, long form of the configuration options – of the MODIFY JOB command.

```
-modify ddname
    [-jobid jobid]
    [-start
        [-immediate]
        [-targetserver server]
        [-wait
            [-poll seconds]
            [-joblog {yes|no}]
            [-spoollist {yes|no}]
            [-purge]
            [-waitchild {yes|no}]
            [-joblogchild {yes|no|error}]
            [-spoollistchild {yes|no}]
            [-purgechild {yes|no}]
        ]
    ]
```

Figure 3.10 MODIFY JOB Command Line Syntax

MODIFY JOB Command Argument

The MODIFY JOB command can be expressed as either:

- -M Short form
- -modify Long form

The MODIFY JOB command argument, *ddname*, specifies the name of the job definition file that contains the modification information.

See Chapter 5 Job Definition Files for additional information on the variant definition file.

MODIFY JOB Command Options

Option Name	Description
IMMEDIATE_JOB	Causes the job to be started immediately.
JOB_ID	Job ID of an existing SAP job to be modified.
JOB_LOG_CHILD	Controls the printing of job logs for child jobs.
PURGE_CHILD_JOBS	Controls the purging of child jobs.
PURGE_JOB	Purge job that has completed processing from SAP system.
RETURN_JOB_LOG	Specification for whether or not the job's joblog is returned.
RETURN_SPOOL_LIST	Specification for whether or not the spoollists of all job steps are returned.
SPOOL_LIST_CHILD	Controls the printing of spoollists for child jobs.
START_JOB	Starts the newly defined job.
TARGET_SERVER	Server on which the job will run.
USAP_POLL	Length of time to wait between job status calls to the SAP system.
WAIT	Wait for the SAP job to complete processing.
WAIT_FOR_CHILD_JOBS	Controls the monitoring of child jobs.

Table 3.7 MODIFY JOB Command Options

3.3.9 MODIFY VARIANT Command

The MODIFY VARIANT command is used to modify an SAP variant that already exists in an SAP system. A USAP variant definition file is used to specify the modifications.

Note: MODIFY VARIANT requires XBP interface 2.0.

(See Section 2.5 Client Fault Tolerance for information on XBP interface 2.0.)

Variant definition files are used to define new SAP variants and to modify existing SAP variants. The same syntactical rules apply to the variant definition file in both cases.

The parameter values specified in a variant definition file replace existing values in the SAP variant definition. If a parameter is not specified in the variant definition file, no change is made to the corresponding value in the existing SAP variant definition.

See Section 5.10 Variant Definition File for additional information on the variant definition file.

MODIFY VARIANT Command Line Syntax

Figure 3.11, below, illustrates the command line syntax – using the command line, long form of the configuration options – of the MODIFY VARIANT command.

```
-modify ddname
```

Figure 3.11 MODIFY VARIANT Command Line Syntax

MODIFY VARIANT Command Argument

The MODIFY VARIANT command can be expressed as either:

- -M Short form
- -modify Long form

The MODIFY VARIANT command argument, *ddname*, specifies the name of the variant definition file that contains the modification information.

See Section 5.10 Variant Definition File for additional information on the variant definition file.

3.3.10 START JOB Command

The START JOB command starts a currently defined SAP job.

START JOB Command Line Syntax

Figure 3.12, below, illustrates the command line syntax – using the command line, long form of the configuration options – of the START JOB command.

```
-start -jobname jobname -jobid jobid
    [-immediate]
    [-targetserver server]
    [-wait
        [-poll seconds]
        [-joblog {yes|no}]
        [-spoollist {yes|no}]
        [-purge]
        [-terminatedec exitcode]
        [-finishedec exitcode]
    ]
    [-bdcwait
         [-bdcjobnameptrn pattern]
         [-bdcjobidptrn pattern]
         [-bdcqidptrn pattern]
         [-qtobecreatedec exitcode]
         [-qunprocessedec exitcode]
         [-qinbackgroundec exitcode]
         [-qfinishedec exitcode]
         [-qerrorec exitcode]
    ]
```

Figure 3.12 START JOB Command Line Syntax

START JOB Command Options

Option Name	Description
BATCH_MONITOR	Causes USAP to perform batch input monitoring for the job being started.
EXIT_JOB_FINISHED	USAP exit code for the SAP job finished status.
EXIT_JOB_TERMINATED	USAP exit code for the SAP job terminated status.
EXIT_QUEUE_BACKGROUND	USAP exit code for the SAP queue state ‘S’ (in background).
EXIT_QUEUE_CREATED	USAP exit code for the SAP queue state ‘C’ (to be created).
EXIT_QUEUE_ERROR	USAP exit code for the SAP queue state ‘E’ (error).
EXIT_QUEUE_FINISHED	USAP exit code for the SAP queue state ‘F’ (finished).
EXIT_QUEUE_UNPROCESSED	USAP exit code for the SAP queue state ‘ ’ (unprocessed).
IMMEDIATE_JOB	Causes the job to be started immediately.
JOB_ID	Job ID of an existing SAP job to use as a model for the new job definition.
JOB_LOG_CHILD	Controls the printing of job logs for child jobs.
JOB_NAME	Name of an existing SAP job to use as a model for the new job definition.
JOB_ID_PATTERN	Locates the header record and determines the offset of the job ID in the RSBDCSUB batch input processing report.
JOB_NAME_PATTERN	Locates the header record and determines the offset of the job name in the RSBDCSUB batch input processing report.
PURGE_CHILD_JOBS	Controls the purging of child jobs.
PURGE_JOB	Purge job that has completed processing from SAP system.
QUEUE_ID_PATTERN	Locates the header record and determines the offset of the queue ID in the RSBDCSUB batch input processing report.
RETURN_JOB_LOG	Specification for whether or not the job’s joblog is returned.
RETURN_SPOOL_LIST	Specification for whether or not the spoollists of all job steps are returned.
SPOOL_LIST_CHILD	Controls the printing of spoollists for child jobs.
TARGET_SERVER	Server on which the job will run.
USAP_POLL	Length of time to wait between job status calls to the SAP system.
WAIT	Wait for the SAP job to complete processing.
WAIT_FOR_CHILD_JOBS	Controls the monitoring of child jobs.

Table 3.8 START JOB Command Options

3.3.11 START FS JOBNET Command

The START FS JOBNET command starts a specified FS job network on an SAP system.

START FS JOBNET Command Line Syntax

Figure 3.13, below, illustrates the command line syntax – using the command line, long form of the configuration options – of the START FS JOBNET command.

```
-start -jnetid jobnet_id -jnetprcid jobnet_process_id
    [-wait
        [-poll seconds]
        [-purge]
        ]
```

Figure 3.13 START FS JOBNET Command Line Syntax

START FS JOBNET Command Options

Option Name	Description
JOB_NETWORK_ID	Network identifier for the pre-existing SAP FS job network being started.
JOB_PROCESS_ID	Process ID of an existing SAP FS job network process to start.
PURGE_JOB	Purge job that has completed processing from SAP system.
USAP_POLL	Length of time to wait between job status calls to the SAP system.
WAIT	Wait for the SAP job to complete processing.

Table 3.9 START FS JOBNET Command Options

3.3.12 WAIT for JOB Command

The WAIT for JOB command allows USAP to reconnect to a started job and monitor it through completion.

WAIT for JOB Command Line Syntax

Figure 3.14, below, illustrates the command line syntax – using the command line, long form of the configuration options – of the WAIT for JOB command.

```
-wait -jobname jobname -jobid jobid
    [-job_stat_check_interval seconds]
    [-joblog {yes|no}]
    [-joblog {yes|no}]
    [-printapprc {yes|no}]
    [-useapprc {yes|no}]
    [-server_stop_conditions codes]
    [-spoollist {yes|no}]
    [-spool_codepage codepage]
    [-transtab translation_table]
    [-terminatedec exitcode]
    [-finishedec exitcode]
    [-poll seconds]
    [-purge]
    [-syslog {yes|no}
        [-syslogpre seconds]
        [-syslogpost seconds]
    ]
    [-waitchild {yes|no}]
    [-max_child_depth depth]
    [-joblogchild {yes|no|error}]
    [-spoollistchild {yes|no}]
    [-purgechild {yes|no}]
    [-max_log_size size]
    [-max_spool_size size]
```

Figure 3.14 WAIT for JOB Command Line Syntax

WAIT for JOB Command Options

Option Name	Description
WAIT	Causes USAP to wait for the SAP job to complete processing
EXIT_JOB_FINISHED	USAP exit code for the SAP job finished status.
EXIT_JOB_TERMINATED	USAP exit code for the SAP job terminated status.
JOB_ID	Job ID of an existing SAP job to use as a model for the new job definition.
JOB_LOG_CHILD	Controls the printing of job logs for child jobs.
JOB_NAME	Name of an existing SAP job to use as a model for the new job definition.
MAX_CHILD_DEPTH	Controls the maximum relationship depth that will be monitored by USAP.
MAX_JOB_LOG_SIZE	Maximum size for job logs.
MAX_SPOOL_LIST_SIZE	Maximum size for spool lists.
PURGE_CHILD_JOBS	Controls the purging of child jobs.
PURGE_JOB	Purge job that has completed processing from SAP system.
RETURN_APPLICATION_LOG	Specification for whether or not the job's application log is returned.
RETURN_APPLICATION_RC	Specification for whether or not the job's application return codes are returned.
RETURN_JOB_LOG	Specification for whether or not the job's joblog is returned.
RETURN_SPOOL_LIST	Specification for whether or not the spoollists of all job steps are returned.
SERVER_STOP_CONDITIONS	Exit code(s) of the executing Universal Connector process that should trigger the locally running Universal Broker to cancel the corresponding SAP job.
SPOOL_CODEPAGE	Codepage used for transferring spool lists from SAP system.
SPOOL_LIST_CHILD	Controls the printing of spoollists for child jobs.
STATUS_CHECK_INTERVAL	Length of time that can elapse, without a change in job status, before a call will be made to synchronize the actual job status with the SAP stored status.
SYSLOG	Specification for whether or not a syslog report is generated on standard error if the job does not complete successfully.
SYSLOG_POST_TIME	Length of time to add to the job **`end`** time when calculating the **`to`** time for the syslog report.
SYSLOG_PRE_TIME	Length of time to subtract from the job release time when calculating the **`from`** time for the syslog report.
TRANSLATION_TABLE	Spoollist translation table file to use for formatting returned spoollists.
USAP_POLL	Length of time to wait between job status calls to the SAP system.
USE_APPLICATION_RC	Specification for whether or not the job's application return codes are used to determine the exit code of the USAP job.
WAIT_FOR_CHILD_JOBS	Controls the monitoring of child jobs.

Table 3.10 WAIT for JOB Command Options

3.3.13 MASS ACTIVITY WAIT Command

The MASS ACTIVITY WAIT command allows USAP to wait for (or reconnect and wait for) a started mass activity job and monitor it, and all its interval jobs, through completiont.

MASS ACTIVITY WAIT Command Line Syntax

Figure 3.15, below, illustrates the command line syntax – using the command line, long form of the configuration options – of the MASS ACTIVITY WAIT command.

```
-mawait -jobname jobname -jobid jobid
    [-poll seconds]
    [-job_stat_check_interval seconds]
    [-joblog {yes|no}]
    [-applog {yes|no}]
    [-printapprc {yes|no}]
    [-useapprc {yes|no}]
    [-transtab translation_table]
    [-purge]
    [-syslog {yes|no}
        [-syslogpre seconds]
        [-syslogpost seconds]
    ]
    [-terminatedec exitcode]
    [-finishedec exitcode]
```

Figure 3.15 MASS ACTIVITY WAIT Command Line Syntax

MASS ACTIVITY WAIT Command Options

Option Name	Description
MASS_ACTIVITY_WAIT	Causes USAP to wait for the SAP mass activity jobs to complete processing.
EXIT_JOB_FINISHED	USAP exit code for the SAP job finished status.
EXIT_JOB_TERMINATED	USAP exit code for the SAP job terminated status.
JOB_ID	Job ID of an existing SAP job to use as a model for the new job definition.
JOB_NAME	Name of an existing SAP job to use as a model for the new job definition.
PURGE_JOB	Purge job that has completed processing from SAP system.
RETURN_APPLICATION_LOG	Specification for whether or not the job's application log is returned.
RETURN_APPLICATION_RC	Specification for whether or not the job's application return codes are returned.
RETURN_JOB_LOG	Specification for whether or not the job's joblog is returned.
STATUS_CHECK_INTERVAL	Length of time that can elapse, without a change in job status, before a call will be made to synchronize the actual job status with the SAP stored status.
SYSLOG	Specification for whether or not a syslog report is generated on standard error if the job does not complete successfully.
SYSLOG_POST_TIME	Length of time to add to the job **end** time when calculating the **to** time for the syslog report.
SYSLOG_PRE_TIME	Length of time to subtract from the job release time when calculating the **from** time for the syslog report.
TRANSLATION_TABLE	Spoollist translation table file to use for formatting returned spoollists.
USAP_POLL	Length of time to wait between job status calls to the SAP system.
USE_APPLICATION_RC	Specification for whether or not the job's application return codes are used to determine the exit code of the USAP job.

Table 3.11 MASS ACTIVITY WAIT Command Options

3.3.14 WAIT for FS JOB NETWORK Command

The WAIT for FS JOB NETWORK command allows USAP to reconnect to a started FS job network and monitor it through completion.

WAIT for FS JOB NETWORK Command Line Syntax

Figure 3.16, below, illustrates the command line syntax – using the command line, long form of the configuration options – of the WAIT for FS JOB NETWORK command.

```
-wait -jnetid jobnetid -jnetprcid processid
      [-poll seconds]
    [-purge]
    [-syslog {yes|no}
        [-syslogpre seconds]
        [-syslogpost seconds]
    ]
    [-max_log_size size]
    [-max_spool_size size]
```

Figure 3.16 WAIT for FS JOB NETWORK Command Line Syntax

WAIT for FS JOB NETWORK Command Options

Option Name	Description
WAIT	Causes USAP to wait for the SAP job network to complete processing.
JOB_NETWORK_ID	Network identifier for the pre-existing SAP FS job network being started.
JOB_PROCESS_ID	Process ID of an existing SAP FS job network process to start.
MAX_JOB_LOG_SIZE	Maximum size for job logs.
MAX_SPOOL_LIST_SIZE	Maximum size for spool lists.
PURGE_JOB	Purge job that has completed processing from SAP system.
SYSLOG	Specification for whether or not a syslog report is generated on standard error if the job does not complete successfully.
SYSLOG_POST_TIME	Length of time to add to the job end time when calculating the **to** time for the syslog report.
SYSLOG_PRE_TIME	Length of time to subtract from the job release time when calculating the **from** time for the syslog report.
USAP_POLL	Length of time to wait between job status calls to the SAP system.

Table 3.12 WAIT for FS JOB NETWORK Command Options

3.3.15 BDCWAIT Command

The BDCWAIT command allows USAP to reconnect to a started batch input processing job and monitor it, and all its generated session processing jobs, through completion.

BDCWAIT Command Line Syntax

Figure 3.17, below, illustrates the command line syntax – using the command line, long form of the configuration options – of the BDCWAIT command.

```
-bdcwait -jobname jobname -jobid jobid
    [-poll seconds]
    [-job_stat_check_interval seconds]
    [-joblog {yes|no}]
    [-applog {yes|no}]
    [-printapprc {yes|no}]
    [-useapprc {yes|no}]
    [-transtab translation_table]
    [-purge]
    [-purge_bdc_map {yes|no}]
    [-syslog {yes|no}
        [-syslogpre seconds]
        [-syslogpost seconds]
    ]
    [-terminatedec exitcode]
    [-finishedec exitcode]
    [-qtobecreatedec exitcode]
    [-qunprocessedec exitcode]
    [-qinbackgroundec exitcode]
    [-qfinishedec exitcode]
    [-qerrorec exitcode]
    [-bdcjobnameptrn pattern]
    [-bdcjobidptrn pattern]
    [-bdcqidptrn pattern]
```

Figure 3.17 BDCWAIT Command Line Syntax

BDCWAIT Command Options

Option Name	Description
BATCH_MONITOR	Causes USAP to perform batch input monitoring for the job being started.
EXIT_JOB_FINISHED	USAP exit code for the SAP job finished status.
EXIT_JOB_TERMINATED	USAP exit code for the SAP job terminated status.
EXIT_QUEUE_BACKGROUND	USAP exit code for the SAP queue state ‘S’ (in background).
EXIT_QUEUE_CREATED	USAP exit code for the SAP queue state ‘C’ (to be created).
EXIT_QUEUE_ERROR	USAP exit code for the SAP queue state ‘E’ (error).
EXIT_QUEUE_FINISHED	USAP exit code for the SAP queue state ‘F’ (finished).
EXIT_QUEUE_UNPROCESSED	USAP exit code for the SAP queue state ‘ ’ (unprocessed).
JOB_ID	Job ID of an existing SAP job to use as a model for the new job definition.
JOB_NAME	Name of an existing SAP job to use as a model for the new job definition.
JOB_ID_PATTERN	Locates the header record and determines the offset of the job ID in the RSBDCSUB batch input processing report.
JOB_NAME_PATTERN	Locates the header record and determines the offset of the job name in the RSBDCSUB batch input processing report.
PURGE_BDC_MAP	Specification for whether or not to delete BDC Batch input session queues that have been processed successfully.
PURGE_JOB	Purge job that has completed processing from SAP system.
QUEUE_ID_PATTERN	Locates the header record and determines the offset of the queue ID in the **RSBDCSUB** batch input processing report.
RETURN_APPLICATION_LOG	Specification for whether or not the job’s application log is returned.
RETURN_APPLICATION_RC	Specification for whether or not the job’s application return codes are returned.
RETURN_JOB_LOG	Specification for whether or not the job’s joblog is returned.
STATUS_CHECK_INTERVAL	Length of time that can elapse, without a change in job status, before a call will be made to synchronize the actual job status with the SAP stored status.
SYSLOG	Specification for whether or not a syslog report is generated on standard error if the job does not complete successfully.
SYSLOG_POST_TIME	Length of time to add to the job end time when calculating the **to** time for the syslog report.
SYSLOG_PRE_TIME	Length of time to subtract from the job release time when calculating the **from** time for the syslog report.
TRANSLATION_TABLE	Spoollist translation table file to use for formatting returned spoollists.
USAP_POLL	Length of time to wait between job status calls to the SAP system.
USE_APPLICATION_RC	Specification for whether or not the job’s application return codes are used to determine the exit code of the USAP job.

Table 3.13 BDCWAIT Command Options

3.3.16 ABORT Command

The ABORT command cancels a running SAP job.

ABORT Command Line Syntax

Figure 3.18, below, illustrates the command line syntax – using the command line, long form of the configuration options – of the ABORT command.

```
-abort –jobname jobname –jobid jobid
```

Figure 3.18 ABORT Command Line Syntax

ABORT Command Options

Option Name	Description
JOB_ID	Job ID of an existing SAP job to use as a model for the new job definition.
JOB_NAME	Name of an existing SAP job to use as a model for the new job definition.

Table 3.14 ABORT Command Options

3.3.17 PURGE JOB Command

The PURGE JOB command deletes a defined SAP job, its joblog, and all of its spoollists.

This command is not available on SAP 3.1 and SAP 4.0 systems.

PURGE JOB Command Line Syntax

Figure 3.19, below, illustrates the command line syntax – using the command line, long form of the configuration options – of the PURGE JOB command.

```
-purge -jobname jobname -jobid jobid
```

Figure 3.19 PURGE JOB Command Line Syntax

PURGE JOB Command Options

Option Name	Description
JOB_ID	Job ID of an existing SAP job to use as a model for the new job definition.
JOB_NAME	Name of an existing SAP job to use as a model for the new job definition.

Table 3.15 PURGE JOB Command Options

3.3.18 PURGE FS JOB NETWORK Command

The PURGE FS JOB NETWORK command deletes a defined SAP FS job network.

PURGE FS JOB NETWORK Command Line Syntax

Figure 3.20, below, illustrates the command line syntax – using the command line, long form of the configuration options – of the PURGE FS JOB NETWORK command.

```
-purge -jnetid jobnetid -jnetprcid processid
```

Figure 3.20 PURGE FS JOB NETWORK Command Line Syntax

PURGE FS JOB NETWORK Command Options

Option Name	Description
JOB_NETWORK_ID	Network identifier for the pre-existing SAP FS job network being started.
JOB_PROCESS_ID	Process ID of an existing SAP FS job network process to start.

Table 3.16 PURGE FS JOB NETWORK Command Options

3.3.19 RAISE EVENT Command

The RAISE EVENT command raises the specified SAP background processing event.

RAISE EVENT Command Line Syntax

Figure 3.21, below, illustrates the command line syntax – using the command line, long form of the configuration options – of the RAISE EVENT command.

```
-raise_bp_event –event_id id –event_parm parm
```

Figure 3.21 RAISE EVENT Command Line Syntax

RAISE EVENT Command Options

Option Name	Description
EVENT_ID	Name of the event.
EVENT_PARAMETER	Optional parameter value for the event.

Table 3.17 RAISE EVENT Command Options

3.3.20 SYNTAX Command

The SYNTAX command checks the syntax of a USAP definition file.

SYNTAX Command Line Syntax

Figure 3.22, below, illustrates the command line syntax – using the command line, long form of the configuration options – of the SYNTAX command.

```
-syntax ddname
```

Figure 3.22 SYNTAX Command Line Syntax

SYNTAX Command Argument

The SYNTAX command can be expressed as either:

- -X Short form
- -syntax Long form

The SYNTAX command argument, *ddname*, specifies the name of the definition file that contains the job, variant, or FS job network definition.

- See Chapter 5 Job Definition Files for additional information on the job definition file.
- See Section 5.10 Variant Definition File for additional information on the variant definition file.
- See Section 5.12 FS Job Network Definition File for additional information on the FS Job Network definition file.

3.3.21 DISPLAY JOBLOG Command

The DISPLAY JOBLOG command displays the job log for a specified SAP job.

DISPLAY JOBLOG Command Line Syntax

Figure 3.23, below, illustrates the command line syntax – using the command line, long form of the configuration options – of the DISPLAY JOBLOG command.

```
-display joblog -jobname jobname -jobid jobid -max_log_size size
```

Figure 3.23 DISPLAY JOBLOG Command Line Syntax

DISPLAY JOBLOG Command Argument

The DISPLAY JOBLOG command can be expressed as either:

- -D Short form
- -display Long form

The DISPLAY JOBLOG command argument, **joblog**, requests the display of a job's joblog.

DISPLAY JOBLOG Command Options

Option Name	Description
JOB_ID	Job ID of an existing SAP job to use as a model for the new job definition.
JOB_NAME	Name of an existing SAP job to use as a model for the new job definition.
MAX_JOB_LOG_SIZE	Maximum size for job logs.

Table 3.18 DISPLAY JOBLOG Command Options

3.3.22 DISPLAY SPOOLLIST Command

The DISPLAY SPOOLLIST command displays the spoollist for a job step.

DISPLAY SPOOLLIST Command Line Syntax

Figure 3.24, below, illustrates the command line syntax – using the command line, long form of the configuration options – of the DISPLAY SPOOLLIST command.

```
-display spoollist –jobname jobname –jobid jobid
         –stepnum stepnumber -max_spool_size size
         [-spool_codepage codepage]
         [–transtab translation_table]
```

Figure 3.24 DISPLAY SPOOLLIST Command Line Syntax

DISPLAY SPOOLLIST Command Argument

The DISPLAY SPOOLLIST command can be expressed as either:

- -D Short form
- -display Long form

The DISPLAY SPOOLLIST command argument, **spoollist**, requests the display of a job step's spoollist.

DISPLAY SPOOLLIST Command Options

Option Name	Description
JOB_ID	Job ID of an existing SAP job to use as a model for the new job definition.
JOB_NAME	Name of an existing SAP job to use as a model for the new job definition.
MAX_SPOOL_LIST_SIZE	Maximum size for spool lists.
SPOOL_CODEPAGE	Codepage used for transferring spool lists from SAP system.
STEP_NUMBER	Step number of the SAP job step.
TRANSLATION_TABLE	spoollist translation table file to use for formatting returned spoollists.

Table 3.19 DISPLAY SPOOLLIST Command Options

3.3.23 DISPLAY STATUS Command

The DISPLAY STATUS command displays the current status for an SAP job. The status is printed to standard output and the exit code of **usap** indicates the status. See Section 3.5 Exit Codes for a complete list of job status exit codes.

DISPLAY STATUS Command Line Syntax

Figure 3.25, below, illustrates the command line syntax – using the command line, long form of the configuration options – of the DISPLAY STATUS command.

```
-display status -jobname jobname -jobid jobid
    [-activeec exitcode     ]
    [-readyec exitcode      ]
    [-scheduledec exitcode  ]
    [-releasedec exitcode   ]
    [-terminatedec exitcode ]
    [-finishedec exitcode   ]
```

Figure 3.25 DISPLAY STATUS Command Line Syntax

DISPLAY STATUS Command Argument

The DISPLAY STATUS command can be expressed as either:

- -D Short form
- -display Long form

The DISPLAY STATUS command argument, **status**, requests a job status.

The status is printed to standard output and the exit code of USAP indicates the status. See Section 3.5 Exit Codes for a complete list of job status exit codes.

DISPLAY STATUS Command Options

Option Name	Description
EXIT_JOB_ACTIVE	USAP exit code for the SAP job **`active`** status.
EXIT_JOB_FINISHED	USAP exit code for the SAP job **`finished`** status.
EXIT_JOB_READY	USAP exit code for the SAP job **`ready`** status.
EXIT_JOB_RELEASED	USAP exit code for the SAP job **`released`** status.
EXIT_JOB_SCHEDULED	USAP exit code for the SAP job **`scheduled`** status.
EXIT_JOB_TERMINATED	USAP exit code for the SAP job **`terminated`** status.
JOB_ID	Job ID of an existing SAP job to use as a model for the new job definition.
JOB_NAME	Name of an existing SAP job to use as a model for the new job definition.

Table 3.20 DISPLAY STATUS Command Options

3.3.24 DISPLAY VARIANTS Command

The DISPLAY VARIANTS command displays the variants available for the specified ABAP program.

DISPLAY VARIANTS Command Line Syntax

Figure 3.26, below, illustrates the command line syntax – using the command line, long form of the configuration options – of the DISPLAY VARIANTS command.

```
-display variants -abapname abapname -varselopt {A|B}
```

Figure 3.26 DISPLAY VARIANTS Command Line Syntax

DISPLAY VARIANTS Command Argument

The DISPLAY VARIANTS command can be expressed as either:

- -D Short form
- -display Long form

The DISPLAY VARIANTS command argument, **variants**, displays the variants defined for ABAP program **abapname**.

- Using **-varselopt** *A* will display the variants that are available for batch and dialog mode.
- Using **-varselopt** *B* will display the variants that are available for batch mode only.

DISPLAY VARIANTS Command Options

Option Name	Description
ABAP_NAME	Name of an ABAP program in an SAP system.
VARIANT_SELECTION	Specification to display either variants available for batch and dialog mode or variants available only for batch mode.

Table 3.21 DISPLAY VARIANTS Command Options

3.3.25 DISPLAY VARIANT Command

The DISPLAY VARIANT command displays the contents of a specified variant.

Note: DISPLAY VARIANT requires XBP interface 2.0.
(See Section 2.5 Client Fault Tolerance for information on XBP interface 2.0.)

DISPLAY VARIANT Command Line Syntax

Figure 3.27, below, illustrates the command line syntax – using the command line, long form of the configuration options – of the DISPLAY VARIANT command.

```
-display variant –variant variantname -varlang language
–abapname abapname
```

Figure 3.27 DISPLAY VARIANT Command Line Syntax

DISPLAY VARIANT Command Argument

The DISPLAY VARIANT command can be expressed as either:

- -D Short form
- -display Long form

The DISPLAY VARIANT command argument, **variant**, displays the specified SAP variant.

DISPLAY VARIANT Command Options

Option Name	Description
ABAP_NAME	Name of an ABAP program in an SAP system.
VARIANT	Pre-existing SAP variant whose contents will be displayed.
VARIANT_LANGUAGE	Preferred language in which to return the variant description.

Table 3.22 DISPLAY VARIANT Command Options

3.3.26 DISPLAY JOBDEF Command

The DISPLAY JOBDEF command displays the definition of the specified SAP job.

DISPLAY JOBDEF Command Line Syntax

Figure 3.28, below, illustrates the command line syntax – using the command line, long form of the configuration options – of the DISPLAY JOBDEF command.

```
-display jobdef -jobname jobname -jobid jobid
```

Figure 3.28 DISPLAY JOBDEF Command Line Syntax

DISPLAY JOBDEF Command Argument

The DISPLAY JOBDEF command can be expressed as either:

- -D Short form
- -display Long form

The DISPLAY JOBDEF command argument, **jobdef**, requests the display of a job's definition.

DISPLAY JOBDEF Command Options

Option Name	Description
JOB_ID	Job ID of an existing SAP job to use as a model for the new job definition.
JOB_NAME	Name of an existing SAP job to use as a model for the new job definition.

Table 3.23 DISPLAY JOBDEF Command Options

3.3.27 DISPLAY SELECT Command

The DISPLAY SELECT command displays a variety of attributes for a list of SAP jobs that match the specified criteria.

DISPLAY SELECT Command Line Syntax

Figure 3.29, below, illustrates the command line syntax – using the command line, long form of the configuration options – of the DISPLAY SELECT command.

```
-display select -jobname jobmask
    [-jobid idmask             ]
    [-selusername userid       ]
    [-fromdate date            ]
    [-todate date              ]
    [-fromtime time            ]
    [-totime time              ]
    [-nodate {yes|no}          ]
    [-withpred {yes|no}        ]
    [-released {yes|no}        ]
    [-scheduled {yes|no}       ]
    [-ready {yes|no}           ]
    [-running {yes|no}         ]
    [-finished {yes|no}        ]
    [-aborted {yes|no}         ]
    [-output output-field-list]
```

Figure 3.29 DISPLAY SELECT Command Line Syntax

DISPLAY SELECT Command Argument

The DISPLAY SELECT command can be expressed as either:

- -D Short form
- -display Long form

The DISPLAY SELECT command argument, **select**, requests the display of all jobs matching the **jobmask** and any additional selection criteria specified. The default output for this command is the job name and job ID for each job found. However, additional fields can be printed using the **–output** option.

Note: This command is not available on SAP 3.1 and SAP 4.0 systems.

DISPLAY SELECT Command Options

Option Name	Description
FROM_DATE	Earliest date to use for job selection or syslog request.
FROM_TIME	Earliest time to use for job selection or syslog request.
JOB_ID	Job ID of an existing SAP job to use as a model for the new job definition.
JOB_NAME	Name of an existing SAP job to use as a model for the new job definition.
NO_START_DATE	Specification for whether or not to include jobs with no start date in selection criteria.
OUTPUT_FIELD_LIST	Additional fields to write for the **`select`** command.
STATUS_ABORTED	Specification for whether or not to include jobs with status aborted in selection criteria.
STATUS_FINISHED	Specification for whether or not to include jobs with status finished in selection criteria.
STATUS_READY	Specification for whether or not to include jobs with status ready in selection criteria.
STATUS_RELEASED	Specification for whether or not to include jobs with status released in selection criteria.
STATUS_RUNNING	Specification for whether or not to include jobs with status running in selection criteria.
STATUS_SCHEDULED	Specification for whether or not to include jobs with status scheduled in selection criteria.
TO_DATE	Latest date to use for job selection or syslog request.
TO_TIME	Latest time to use for job selection or syslog request.
USER_NAME	User ID associated with a job.
WITH_PREDECESSOR	Specification for whether or not to include jobs with start after predecessor in selection criteria.

Table 3.24 DISPLAY SELECT Command Options

3.3.28 DISPLAY SYSLOG Command

The DISPLAY SYSLOG command displays a portion of an SAP syslog that meets the specified date/time constraints.

DISPLAY SYSLOG Command Line Syntax

Figure 3.30, below, illustrates the command line syntax – using the command line, long form of the configuration options – of the DISPLAY SYSLOG command.

```
-display syslog -fromdate date -todate date
    [-fromtime time      ]
    [-totime time        ]
    [-pagelimit limit    ]
    [-targetserver server]
```

Figure 3.30 DISPLAY SYSLOG Command Line Syntax

DISPLAY SYSLOG Command Argument

The DISPLAY SYSLOG command can be expressed as either:

- -D Short form
- -display Long form

The DISPLAY SYSLOG command argument, **syslog**, requests entries from an SAP System syslog for a specified date and time range.

DISPLAY SYSLOG Command Options

Option Name	Description
FROM_DATE	Earliest date to use for job selection or syslog request.
FROM_TIME	Earliest time to use for job selection or syslog request.
PAGE_LIMIT	Maximum number of pages that can be returned in the syslog report.
TARGET_SERVER	Name of the server whose syslog will be read.
TO_DATE	Latest date to use for job selection or syslog request.
TO_TIME	Latest time to use for job selection or syslog request.

Table 3.25 DISPLAY SYSLOG Command Options

3.3.29 DISPLAY INTERCEPTED_JOBS Command

The DISPLAY INTERCEPTED_JOBS command displays intercepted jobs for the connected SAP system.

DISPLAY INTERCEPTED_JOBS Command Line Syntax

Figure 3.31, below, illustrates the command line syntax – using the command line, long form of the configuration options – of the DISPLAY INTERCEPTED_JOBS command.

```
-display intercepted_jobs -dspclient client
```

Figure 3.31 DISPLAY INTERCEPTED_JOBS Command Line Syntax

DISPLAY INTERCEPTED_JOBS Command Argument

The DISPLAY INTERCEPTED_JOBS command can be expressed as either:

- -D Short form
- -display Long form

The DISPLAY INTERCEPTED_JOBS command argument, **intercepted_jobs**, requests the display of an SAP system's intercepted jobs. Unless a specific client is identified, intercepted jobs for all clients are displayed.

DISPLAY INTERCEPTED_JOBS Command Options

Option Name	Description
DISPLAY_CLIENT	Specific SAP client whose intercepted jobs will be reported.

Table 3.26 DISPLAY INTERCEPTED_JOBS Command Options

3.3.30 DISPLAY INTERCEPT_TABLE Command

The DISPLAY INTERCEPT_TABLE command displays the contents of the job intercept criteria table for the connected SAP system.

DISPLAY INTERCEPT_TABLE Command Line Syntax

Figure 3.32, below, illustrates the command line syntax – using the command line, long form of the configuration options – of the DISPLAY INTERCEPT_TABLE command.

```
-display intercept_table
```

Figure 3.32 DISPLAY INTERCEPT_TABLE Command Line Syntax

DISPLAY INTERCEPT_TABLE Command Argument

The DISPLAY INTERCEPT_TABLE command can be expressed as either:

- -D Short form
- -display Long form

The DISPLAY INTERCEPT_TABLE command argument, **intercept_table**, requests the display of an SAP system's job intercept criteria table.

3.3.31 DISPLAY QSTATE Command

The DISPLAY QSTATE command displays the state of a specific Batch Input / BDC session queue in an SAP system.

DISPLAY QSTATE Command Line Syntax

Figure 3.33, below, illustrates the command line syntax – using the command line, long form of the configuration options – of the DISPLAY QSTATE command.

```
-display qstate -qid queueid
```

Figure 3.33 DISPLAY QSTATE Command Line Syntax

DISPLAY QSTATE Command Argument

The DISPLAY QSTATE command can be expressed as either:

- -D Short form
- -display Long form

The DISPLAY QSTATE command argument, **qstate**, requests the state of a queue used to process a batch input session. See Section 3.5.4 DISPLAY QSTATE Exit Codes for a complete list of queue state exit codes.

DISPLAY QSTATE Command Options

Option Name	Description
QUEUE_ID	Queue identifier associated with the batch input session.

Table 3.27 DISPLAY QSTATE Command Options

3.3.32 DISPLAY REPORTS Command

The DISPLAY REPORTS command displays a list of ABAP reports that match the specified criteria.

DISPLAY REPORTS Command Line Syntax

Figure 3.34, below, illustrates the command line syntax – using the command line, long form of the configuration options – of the DISPLAY REPORTS command.

```
-display reports -abapname abapmask -count max_hit_count
```

Figure 3.34 DISPLAY REPORTS Command Line Syntax

DISPLAY REPORTS Command Argument

The DISPLAY REPORTS command can be expressed as either:

- -D Short form
- -display Long form

The DISPLAY REPORTS command argument, **reports**, requests the display of a list of ABAP reports that match the specified criteria.

DISPLAY REPORTS Command Options

Option Name	Description
ABAP_NAME	Complete ABAP name or a mask used to select SAP ABAP reports that match the mask.
MAX_HIT_COUNT	Maximum number of ABAP reports to be returned.

Table 3.28 DISPLAY REPORTS Command Options

3.3.33 DISPLAY COMMANDS Command

The DISPLAY COMMANDS command displays a list of SAP external commands that match the specified criteria.

DISPLAY COMMANDS Command Line Syntax

Figure 3.35, below, illustrates the command line syntax – using the command line, long form of the configuration options – of the DISPLAY COMMANDS command.

```
-display commands -cmd external_command_mask -opsys operating_system
```

Figure 3.35 DISPLAY COMMANDS Command Line Syntax

DISPLAY COMMANDS Command Argument

The DISPLAY COMMANDS command can be expressed as either:

- -D Short form
- -display Long form

The DISPLAY COMMANDS command argument, **commands**, requests the display of a list of SAP external commands that match the specified criteria.

DISPLAY COMMANDS Command Options

Option Name	Description
EXTERNAL_COMMAND	Complete command name or a mask used to select SAP external commands that match the mask.
OPERATING_SYSTEM	Name of the operating system for which external commands are searched.

Table 3.29 DISPLAY COMMANDS Command Options

3.3.34 DISPLAY OUTPUT_DEVICES Command

The DISPLAY OUTPUT_DEVICES command displays a list of SAP output devices that match the specified criteria.

DISPLAY OUTPUT_DEVICES Command Line Syntax

Figure 3.36, below, illustrates the command line syntax – using the command line, long form of the configuration options – of the DISPLAY OUTPUT_DEVICES command.

```
-display output_devices -short_name technical_device_name_mask
    -long_name long_device_name_mask
```

Figure 3.36 DISPLAY OUTPUT_DEVICES Command Line Syntax

DISPLAY OUTPUT_DEVICES Command Argument

The DISPLAY OUTPUT_DEVICES command can be expressed as either:

- -D Short form
- -display Long form

The DISPLAY OUTPUT_DEVICES command argument, **output_devices**, requests the display of a list of SAP output devices that match the specified criteria.

DISPLAY OUTPUT_DEVICES Command Options

Option Name	Description
TECHNICAL_DEVICE_NAME	Complete device name or a mask used to select SAP output devices that match the mask.
LONG_DEVICE_NAME	Complete device name or a mask used to select SAP output devices that match the mask.

Table 3.30 DISPLAY OUTPUT_DEVICES Command Options

3.3.35 DISPLAY PRINT_FORMATS Command

The DISPLAY PRINT_FORMATS command displays a list of print formats that are available for the specified printer.

DISPLAY PRINT_FORMATS Command Line Syntax

Figure 3.37, below, illustrates the command line syntax – using the command line, long form of the configuration options – of the DISPLAY PRINT_FORMATS command.

```
-display print_formats -printer printer_name -layout layout
```

Figure 3.37 DISPLAY PRINT_FORMATS Command Line Syntax

DISPLAY PRINT_FORMAT Command Argument

The DISPLAY PRINT_FORMAT command can be expressed as either:

- -D Short form
- -display Long form

The DISPLAY PRINT_FORMAT command argument, **print_formats**, requests the display of a list of print formats available for the specified printer.

DISPLAY PRINT_FORMATS Command Options

Option Name	Description
LAYOUT_NAME	Complete layout name or a mask used to select printer layouts that match the mask.
PRINTER_NAME	Name of a printer for which the print formats will be retrieved.

Table 3.31 DISPLAY PRINT_FORMATS Command Options

3.3.36 GENERATE JOB DEFINITION FILE Command

The GENERATE JOB DEFINITION FILE command generates a USAP job definition file based on a model SAP job. The generated definition file is written to standard output.

GENERATE JOB DEFINITION FILE Command Line Syntax

Figure 3.38, below, illustrates the command line syntax – using the command line, long form of the configuration options – of the GENERATE JOB DEFINITION FILE command.

```
-generate jobdef -jobname jobname -jobid jobid
```

Figure 3.38 GENERATE JOB DEFINITION FILE Command Line Syntax

GENERATE JOB DEFINITION FILE Command Argument

The GENERATE JOB DEFINITION FILE command can be expressed as:

- -generate Long form

The GENERATE JOB DEFINITION FILE command argument, ***jobdef***, generates a usap job definition file based on the specified SAP job definition. The generated job definition is printed to standard output.

This command option makes it easy to create complex job definitions based on pre-existing SAP jobs. It also eliminates typing errors that can be introduced by manually coding job definition files.

GENERATE JOB DEFINITION FILE Command Options

Option Name	Description
JOB_ID	Job ID of an existing SAP job to select as the model job.
JOB_NAME	Name of an existing SAP job to select as the model job.

Table 3.32 GENERATE JOB DEFINITION FILE Command Options

3.3.37 GENERATE VARIANT DEFINITION FILE Command

The GENERATE VARIANT DEFINITION FILE command generates a USAP variant definition file based on a model SAP variant. The generated definition file is written to standard output.

Note: GENERATE VARIANT DEFINITION FILE requires XBP interface 2.0.

(See Section 2.5 Client Fault Tolerance for information on XBP interface 2.0.)

GENERATE VARIANT DEFINITION FILE Command Line Syntax

Figure 3.39, below, illustrates the command line syntax – using the command line, long form of the configuration options – of the GENERATE VARIANT DEFINITION FILE command.

```
-generate vardef -variant variantname -abapname abapname
```

Figure 3.39 GENERATE VARIANT DEFINITION FILE Command Line Syntax

GENERATE VARIANT DEFINITION FILE Command Argument

The GENERATE VARIANT DEFINITION FILE command can be expressed as:

- -generate Long form

The GENERATE VARIANT DEFINITION FILE command argument, **vardef**, generates a usap variant definition file based on the specified SAP variant. The generated variant definition is printed to standard output.

This command option makes it easy to create complex variant definitions based on pre-existing SAP variants. It also eliminates typing errors that can be introduced by manually coding variant definition files.

GENERATE VARIANT DEFINITION FILE Command Options

Option Name	Description
VARIANT	Pre-existing SAP variant name to use as the model variant.
ABAP_NAME	Name of an ABAP program in an SAP system to which the model variant belongs.

Table 3.33 GENERATE VARIANT DEFINITION FILE Command Options

3.4 Configuration Options

This section identifies the Universal Connector for z/OS configuration options that are not associated with one or more Universal Connector commands. Some of these options are required for every execution of Universal Connector; others are optional for any execution.

(For detailed information on configuration options that are associated with one or more specific commands, see Section 3.3 Commands.)

3.4.1 Configuration Option Categories

The configuration options not associated with one or more specific commands are categorized into logical areas of application, as shown in Table 3.34, below.

The name of each category is a link to the following information in this section:

- Description — Description of the options in the category.
- Options syntax — Syntax of the options on the command line.
- Options — Description of the configuration options in the category and a link to detailed information about those options in the Universal Connector 4.1.0 Reference Guide.

Option Categories	Description
Required	
HOST Options	Specifies the SAP host to which a connection should be made.
USER Options	Identifies the SAP user account with which the command executes.
Optional	
CFT Options	Configures client fault tolerant connection.
COMMAND FILE Options	Specifies an additional source of command options.
EVENT Options	Specifies USAP options required for event generation.
Exit Codes	Requests information pertaining to the USAP program.
LOCAL Options	Specifies USAP options required for local broker registration.
MESSAGE Options	Requests information pertaining to the USAP program.
RFC Options	Configures fault tolerant RFC connection.

Table 3.34 Universal Connector for z/OS - Configuration Option Categories

3.4.2 HOST Options

The HOST options are required to establish a connection with an SAP system.

HOST Options Syntax

Figure 3.40, below, illustrates the command line syntax – using the command line, long form of the configuration options – of the HOST options.

```
{-dest destination | -ashost hostname -sysnr number}
-client client
[-max_xbp version]
[-saplang language]
[-xmiaudit {0|1|2|3}]
```

Figure 3.40 HOST Options Syntax

HOST Options

Option Name	Description
AS_HOST	Host name of an SAP application server.
DESTINATION	Name of a destination defined in the **saprfc.ini** file.
CLIENT	SAP client number.
LOGON_LANGUAGE	SAP logon language used for the USAP session.
MAX_XBP	Maximum version of the SAP XBP interface that will be used during USAP execution.
SYSTEM_NUMBER	SAP system number of an SAP application server.
XMI_AUDIT_LEVEL	Sets the XMI audit level to be used for the execution of the command.

Table 3.35 HOST Options

3.4.3 USER Options

The USER options are required to establish an RFC connection to an SAP system. They establish the SAP user identity.

USER Options Syntax

Figure 3.41, below, illustrates the command line syntax – using the command line, long form of the configuration options – of the USER options.

```
-userid userid
-pwd password
```

Figure 3.41 USER Options Syntax

USER Options

Option Name	Description
USER_ID	SAP user ID with which to logon to the SAP system.
PASSWORD	Password for the SAP user ID.

Table 3.36 USER Options

3.4.4 CFT Options

The CFT options are used to configure a client fault tolerant job run. Client fault tolerance is requested for a USAP job run by specifying a COMMAND_ID.

CFT Options Syntax

Figure 3.42, below, illustrates the command line syntax – using the command line, long form of the configuration options – of the CFT options.

```
-cmdid id
-restart {yes|no|auto}
-autorestartok {yes|no}
-cft_secure_cft {yes|no}
-cft_abap abap_program
-cft_target_host host
-cft_cmd_prefix command_prefix
```

Figure 3.42 CFT Options Syntax

CFT Options

Option Name	Description
ALLOW_AUTO_RESTART	Specification for whether or not a **RESTART** value of **AUTO** will be allowed.
CFT_ABAP_PROGRAM	ABAP program to use for the command ID job step.
CFT_COMMAND_PREFIX	In pre-XBP 2.0 CFT mode, the prefix command required for the operating system of the target host.
CFT_TARGET_HOST	In pre-XBP 2.0 CFT mode, the target host to use for the command ID job step when the command ID option is used.
COMMAND_ID	Identifier used to identify the unit of work represented by a USAP command and the associated SAP job.
RESTART	Specification for whether or not this execution of USAP is a restart of a previous client fault tolerant USAP command.
SECURE_CFT	Mode of client fault tolerance to be used for the command invocation.

Table 3.37 CFT Options

3.4.5 COMMAND FILE Options

The COMMAND FILE options are used to specify a file as a source of configuration options used for a command execution. The options read from a command file are processed exactly like options from any other input source.

Encrypted command files are an excellent place to store sensitive data such as user IDs and passwords. Use the Universal Encrypt utility to encrypt a plain text command file. (For information on Universal Encrypt, see the Universal Products Utilities 4.1.0 User Guide.)

Note: All options, including required and command-specific options, can be placed in a command file.

USAP can process both plain text and encrypted command files. Either type of file can be used, but not both. If both are specified, the plain text file will be used.

COMMAND FILE (Plain Text) Options Syntax

Figure 3.43, below, illustrates the command line syntax – using the command line, long form of the configuration options – of the COMMAND FILE (Plain Text) options.

```
-file [ddname]
```

Figure 3.43 Command File (Plain Text) Options Syntax

COMMAND FILE (Plain Text) Options

Option Name	Description
FILE_NAME	Name of a plain text command file.

Table 3.38 COMMAND FILE (Plain Text) Options

COMMAND FILE (Encrypted) Options Syntax

Figure 3.44, below, illustrates the command line syntax – using the command line, long form of the configuration options – of the COMMAND FILE (Encrypted) options.

```
-encryptedfile [ddname [-key key] ]
```

Figure 3.44 Command File (Encrypted) Options Syntax

COMMAND FILE (Encrypted) Options

Option Name	Description
ENCRYPT_FILE	Name of an encrypted command file.
ENCRYPTION_KEY	Key used to encrypt the command file.

Table 3.39 COMMAND FILE (Encrypted) Options

3.4.6 EVENT Options

The EVENT options are required for event generation.

Note: EVENT options can be specified only in the configuration file. They have no command line or environment variable parameters.

EVENT Options Syntax

Figure 3.45, below, illustrates the configuration file syntax of the EVENT options.

```
activity_monitoring {yes|no}
event_generation types
```

Figure 3.45 EVENT Options Syntax

EVENT Options

Option Name	Description
ACTIVITY_MONITORING	Specification for whether or not product activity monitoring events are generated.
EVENT_GENERATION	Events to be generated as persistent events.

Table 3.40 EVENT Options

3.4.7 INFORMATIONAL Options

The INFORMATIONAL options request information pertaining to the USAP program.

INFORMATIONAL Options Syntax

Figure 3.46, below, illustrates the command line syntax – using the command line, long form of the configuration options – of the INFORMATIONAL options.

```
-help
-version
```

Figure 3.46 INFORMATIONAL Options Syntax

INFORMATIONAL Options

Option Name	Description
HELP	Writes command line help.
VERSION	Writes USAP version and copyright information.

Table 3.41 INFORMATIONAL Options

3.4.8 LOCAL Options

The LOCAL options are required for local broker registration.

LOCAL Options Syntax

Figure 3.47, below, illustrates the command line syntax – using the command line, long form of the configuration options – of the LOCAL options.

```
-system_id id
```

Figure 3.47 LOCAL Options Syntax

LOCAL Options

Option Name	Description
SYSTEM_ID	Local Universal Broker with which USAP must register.

Table 3.42 LOCAL Options

3.4.9 MESSAGE Options

The MESSAGE options specify different characteristics of **usap** messages.

MESSAGE Options Syntax

Figure 3.48, below, illustrates the command line syntax – using the command line, long form of the configuration options – of the MESSAGE options.

```
-lang language
-level {trace|audit|info|warn|error}
-trace_file_lines lines
-trace_table size,condition
```

Figure 3.48 MESSAGE Options Syntax

MESSAGE Options

Option Name	Description
MESSAGE_LANGUAGE	Language in which messages are written.
MESSAGE_LEVEL	Level of messages to be written.
TRACE_FILE_LINES	Maximum number of lines to write to the trace file.
TRACE_TABLE	Trace table size and under what conditions it is written to a file.

Table 3.43 MESSAGE Options

3.4.10 RFC Options

The RFC options always are used to configure a fault tolerant RFC connection. All RFC options have default values that are used if additional values are not provided.

RFC Options Syntax

Figure 3.49, below, illustrates the command line syntax – using the command line, long form of the configuration options – of the HOST options.

```
-rfc_logon_retry_interval interval
-rfc_logon_retry_count count
-rfc_listen_interval interval
-rfc_timeout interval
-rfc_retry_interval interval
-rfc_retry_count count
-rfc_trace_dir path
```

Figure 3.49 RFC Options Syntax

RFC Options

Option Name	Description
LOGON_RETRY_INTERVAL	Number of seconds that will elapse between a failed RFC logon attempt and the retry of that logon attempt.
LOGON_RETRY_COUNT	Number of unsuccessful RFC logon retry attempts that can occur before USAP terminates the logon process and ends unsuccessfully.
LISTEN_INTERVAL	Number of seconds that will elapse between RFC listen calls.
TIMEOUT_INTERVAL	Number of seconds that can elapse before USAP considers an RFC call to have timed out.
RETRY_CALL_INTERVAL	Number of seconds that will elapse between a failed RFC call and the retry of that call.
RETRY_CALL_COUNT	Number of unsuccessful RFC call retry attempts that can occur before USAP terminates the RFC call retry process and ends unsuccessfully.
TRACE_DIRECTORY	Directory where RFC trace files will be written.

Table 3.44 RFC Options

3.5 Exit Codes

The USAP exit code depends on the command being issued.

The following sections identify the exit codes for the various USAP commands.

Note: The default values listed for the exit codes are the installed (configuration file) values. These values may be different than the internal default values (see the Exit Code options in the Universal Connector 4.1.0 Reference Guide).

3.5.1 WAIT for JOB Exit Codes

If the WAIT for JOB command is specified, USAP will map the job's status upon completion to the user-definable job exit code parameters.

Table 3.45, below, illustrates this mapping. USAP default values are listed in parentheses.

Job Completion Status in SAP	Exit Code
Terminated	terminated_exit_code (8)
Finished	finished_exit_code (0)
Unknown	22
Error in USAP processing (see Table 3.49).	> 200

Table 3.45 z/OS Wait for Job Exit Codes

3.5.2 WAIT for FS JOB NETWORK Exit Codes

If the WAIT for FS JOB NETWORK command is specified, USAP will map the job network's return code pair to the user-definable job network return code parameters. In this case, the exit codes are hard coded and the return code pairs used in the matching process are user definable.

Table 3.46, below, illustrates this relationship. USAP default values are listed in parentheses.

Job Network Return Code Pairs Used for Matching	Exit Code
job_net_rc_00 (02,00;02,02)	0
job_net_rc_04 (02,02)	4
job_net_rc_08 (02,04)	8
job_net_rc_16 (07,00;04,00;02,08)	16
Error in USAP processing (see Table 3.49).	> 200

Table 3.46 z/OS Wait for FS Job Network Exit Codes

3.5.3 DISPLAY STATUS Exit Codes

If the DISPLAY STATUS command is specified, USAP will map the job's current status to the user definable job exit code parameters.

Table 3.47, below, illustrates this relationship. USAP default values are listed in parentheses.

Job Completion Status in SAP	Exit Code
Active	active_exit_code (10)
Ready	ready_exit_code (12)
Scheduled	scheduled_exit_code (14)
Released	released_exit_code (16)
Terminated	terminated_exit_code (8)
Finished	finished_exit_code (0)
Unknown	22
Error in USAP processing (see Table 3.49).	> 200

Table 3.47 z/OS DISPLAY STATUS Exit Codes

3.5.4 DISPLAY QSTATE Exit Codes

If the DISPLAY QSTATE command is specified, USAP will map the queue's current state to the user definable **`qstate`** exit code parameters.

Table 3.48, below, illustrates this relationship. USAP default values are listed in parentheses.

Queue State	Exit Code
'C' to be created	qtobecreated_exit_code (14)
' ' unprocessed	qunprocessed_exit_code (12)
'S' in background	qinbackground_exit_code (10)
'E' error	qerror_exit_code (8)
'F' finished	qfinished_exit_code (0)
Undefined	20
Error in USAP processing (see Table 3.49).	> 200

Table 3.48 z/OS Queue State Exit Codes

3.5.5 All Other Command Exit Codes

If USAP is not performing the WAIT for JOB, WAIT for FS JOB NETWORK, DISPLAY STATUS, or DISPLAY QSTATE command, the exit code indicates the success of the requested actions.

Table 3.49, below, lists the USAP for z/OS exit codes.

Exit Code	Description
0	Successfully completed all requested actions.
201	An error occurred processing the requested actions. Messages are printed providing details about the error.
210	Indicates an error with product configuration options or command line options.
211	An error occurred in the initialization phase of message processing. It is possible the error prohibited messages from printing.

Table 3.49 USAP for z/OS - Exit Codes

Chapter 4
Universal Connector for UNIX

4.1 Overview

This chapter documents Universal Connector (USAP) at a detailed level, specific to UNIX-based operating systems.

It provides sections of the following information:

- Usage
- Commands
- Configuration Options
- Exit Codes

4.2 Usage

Universal Connector for UNIX executes as a command line application.

Each command line execution contains:

1. Universal Connector command (and argument).
2. Configuration options associated with that command.
3. Configuration options (required and optional) not associated with any specific command.

Each execution of Universal Connector performs an operation specified by the command. The configuration options describe information / actions for that operation.

4.2.1 Command Line Syntax

Figure 4.1, below, illustrates the command line syntax of Universal Connector for UNIX.

```
usap  {RUN | SUBMIT | MODIFY | START | WAIT | ABORT | DISPLAY | GENERATE |
      PURGE | SYNTAX | RAISE EVENT} HOST USER [CFT] [EVENT] [INSTALLATION]
      [INFORMATIONAL] [LOCAL] [MESSAGE] [RFC] [COMMAND FILE]
```

Figure 4.1 Universal Connector for UNIX - Command Line Syntax

Names enclosed in {BRACES} identify command groups. For each execution, a single command is specified from one of these groups. One or more configuration options associated with each command also can be used to specify additional information / actions for the execution.

Names encoded in [BRACKETS] identify categories of configuration options that are not associated with specific commands and from which options are not required.

Names not enclosed in {BRACES} or [BRACKETS] identify categories of configuration options that are not associated with specific commands but from which one or more options are required.

See Section 4.3 Commands for detailed information on the commands, and their associated configuration options, in each command group.

See Section 4.4 Configuration Options for detailed information on configuration options not associated with one or more commands.

Example

The following is an example of a command line syntax executing Universal Connector:

```
usap -sub file.usp -immediate -client 987 -dest -userid 123 -pwd ABC
-lang english -level info
```

4.3 Commands

This section identifies all of the Universal Connector for UNIX commands.

Each command has configuration options associated with it that can be used to specify additional information / actions for an execution of that command.

(For detailed information on configuration options not associated with one or more specific commands, see Section 4.4 Configuration Options.)

4.3.1 Command Groups

Universal Connector groups commands into areas of common functionality, as shown in Table 4.1, below. Each row identifies a command group, the commands in that group, and the type of operation performed by those commands.

The name of each command is a link to the following information in this section:

- Command description — Description of the operation(s) performed by the command.
- Command line syntax — Syntax of the command and its options on the command line.
- Command argument — Command line expression (short and/or long form) and description of the command argument.
- Command options — Description of the configuration options associated with the option and a link to detailed information about those options in the Universal Connector 4.1.0 Reference Guide.

Command Groups	Description
RUN • RUN JOB Command • RUN FS JOB NETWORK Command	Performs the following actions: 1. Defines an SAP job. 2. Starts the job. 3. Waits for the job to complete. 4. Writes the joblog and spoollists of the job. 5. Purges the job from the SAP system.
SUBMIT • SUBMIT FS JOBNET Command • SUBMIT JOB Command • SUBMIT INTERCEPT CRITERIA TABLE Command • SUBMIT VARIANT Command	Defines a job to the SAP system.
MODIFY • MODIFY JOB Command • MODIFY VARIANT Command	Modifies a job in an SAP job.
START • START FS JOBNET Command • START JOB Command	Starts a defined SAP job.

Command Groups	Description
WAIT • WAIT for JOB Command • MASS ACTIVITY WAIT Command • WAIT for FS JOB NETWORK Command • BDCWAIT Command	Allows USAP to reconnect to a started job and monitor it through completion.
ABORT • ABORT Command	Cancels a running SAP job.
DISPLAY • DISPLAY COMMANDS Command • DISPLAY INTERCEPTED_JOBS Command • DISPLAY INTERCEPT_TABLE Command • DISPLAY JOBDEF Command • DISPLAY JOBLOG Command • DISPLAY OUTPUT_DEVICES Command • DISPLAY PRINT_FORMATS Command • DISPLAY QSTATE Command • DISPLAY REPORTS Command • DISPLAY SPOOLLIST Command • DISPLAY STATUS Command • DISPLAY VARIANT Command • DISPLAY VARIANTS Command • DISPLAY SELECT Command • DISPLAY SYSLOG Command	Displays various SAP job data.
GENERATE • GENERATE JOB DEFINITION FILE Command • GENERATE VARIANT DEFINITION FILE Command	Generates USAP job or variant definitions based on model SAP jobs or variants.
PURGE • PURGE JOB Command • PURGE FS JOB NETWORK Command	Deletes SAP jobs.
RAISE EVENT Command	Raises the specified SAP background processing event.
SYNTAX Command	Checks the syntax of a job definition file.

Table 4.1 USAP for UNIX - Command Groups

4.3.2 RUN JOB Command

The RUN JOB command performs the following actions:

1. Defines a new SAP job, based on either a job definition specification or an existing SAP job definition.
2. Starts the defined job.
3. Waits for the job to complete.
4. Writes the job's joblog to standard error and the spoollists to standard output.
5. Purges the job from the SAP system.

The exit code of USAP will indicate the completion status of the SAP job.

See Section 4.5.1 WAIT for JOB Exit Codes for a complete list of job status exit codes.

RUN JOB Command Line Syntax

Figure 4.2, below, illustrates the command line syntax – using the command line, long form of the configuration options – of the RUN JOB command.

```
-run {filename | -jobname jobname –jobid jobid}
    [-target_jobname jobname]
    [-poll seconds]
    [-job_stat_check_interval seconds]
    [-targetserver server]
    [-target_variant job step,variant name;job step,variant name;...]
    [-immediate]
    [-cmopctoken token]
    [-cmdid id]
    [-cmmode {N|R|S|T}]
    [-activeec exitcode]
    [-readyec exitcode]
    [-scheduledec exitcode]
    [-releasedec exitcode]
    [-terminatedec exitcode]
    [-finishedec exitcode]
    [-max_log_size size]
    [-max_spool_size size]
    [-spool_codepage codepage]
    [-bdcwait
         [-bdcjobnameptrn pattern]
         [-bdcjobidptrn pattern]
         [-bdcqidptrn pattern]
         [-qtobecreatedec exitcode]
         [-qunprocessedec exitcode]
         [-qinbackgroundec exitcode]
         [-qfinishedec exitcode]
         [-qerrorec exitcode]
    ]
```

Figure 4.2 RUN JOB Command Line Syntax

RUN JOB Command Argument

The RUN JOB command can be expressed as either:

- -R Short form
- -run Long form

The RUN JOB command argument, *filename*, specifies the name of the file that contains the job definition.

See Chapter 5 Job Definition Files for additional information on the job definition file.

RUN JOB Command Options

Option Name	Description
BATCH_MONITOR	Causes USAP to perform batch input monitoring for the job being started.
CM_ID	Name of the CM communication to be started.
CM_MODE	Starting mode of a CM Communication.
CM_TOKEN	Unique character string used to prefix SAP jobs.
EXIT_JOB_ACTIVE	USAP exit code for the SAP job active status.
EXIT_JOB_FINISHED	USAP exit code for the SAP job finished status.
EXIT_JOB_READY	USAP exit code for the SAP job ready status.
EXIT_JOB_RELEASED	USAP exit code for the SAP job released status.
EXIT_JOB_SCHEDULED	USAP exit code for the SAP job scheduled status.
EXIT_JOB_TERMINATED	USAP exit code for the SAP job terminated status.
EXIT_QUEUE_BACKGROUND	USAP exit code for the SAP queue state **S** (in background).
EXIT_QUEUE_CREATED	USAP exit code for the SAP queue state **C** (to be created).
EXIT_QUEUE_ERROR	USAP exit code for the SAP queue state **E** (error).
EXIT_QUEUE_FINISHED	USAP exit code for the SAP queue state **F** (finished).
EXIT_QUEUE_UNPROCESSED	USAP exit code for the SAP queue state [ ] (unprocessed).
IMMEDIATE_JOB	Causes the job to be started immediately.
JOB_ID	Job ID of an existing SAP job to use as a model for the new job definition.
JOB_NAME	Name of an existing SAP job to use as a model for the new job definition.
JOB_ID_PATTERN	Locates the header record and determines the offset of the job ID in the RSBDCSUB batch input processing report.
JOB_NAME_PATTERN	Locates the header record and determines the offset of the job name in the RSBDCSUB batch input processing report.
MAX_JOB_LOG_SIZE	Maximum size for job logs.
MAX_SPOOL_LIST_SIZE	Maximum size for spool lists.
QUEUE_ID_PATTERN	Locates the header record and determines the offset of the queue ID in the RSBDCSUB batch input processing report.
SPOOL_CODEPAGE	Codepage used for transferring spool lists from SAP system.
STATUS_CHECK_INTERVAL	Length of time that can elapse, without a change in job status, before a call will be made to synchronize the actual job status with the SAP stored status.
TARGET_JOB_NAME	Name to give the newly created job.
TARGET_SERVER	Server on which the job will run.
TARGET_VARIANT	One or more replacement variants for ABAP program job steps in an SAP job.
USAP_POLL	Length of time to wait between job status calls to the SAP system.

Table 4.2 RUN JOB Command Options

4.3.3 RUN FS JOB NETWORK Command

The RUN FS JOB NETWORK command performs the following actions:

1. Defines a new SAP FS job network based on a USAP FS Job Network definition file.
2. Starts the defined FS job network.
3. Waits for the started FS job network to complete.
4. Purges the FS job network from the SAP system.

The exit code of **usap** will indicate the completion status of the FS job network.

See Section 4.5.2 WAIT for FS JOB NETWORK Exit Codes for a complete list of job status exit codes.

RUN FS JOB NETWORK Command Line Syntax

Figure 4.3, below, illustrates the command line syntax – using the command line, long form of the configuration options – of the RUN JOB NETWORK command.

```
-run {filename | -jnetid jobnetid -jnetprcid processid}
```

Figure 4.3 RUN FS JOB NETWORK Command Line Syntax

RUN FS JOB Command Argument

The RUN FS JOB command can be expressed as either:

- -R Short form
- -run Long form

The RUN FS JOB command argument, *filename*, specifies the name of the file that contains the FS job network definition.

See Section 5.12 FS Job Network Definition File for additional information on the FS job network definition file.

RUN FS JOB NETWORK Command Options

Option Name	Description
JOB_NETWORK_ID	Network identifier for the pre-existing SAP FS job network being started.
JOB_PROCESS_ID	Process ID of an existing SAP FS job network process to start.

Table 4.3 RUN FS JOB NETWORK Command Options

4.3.4 SUBMIT JOB Command

The SUBMIT JOB command defines a new SAP job.

SUBMIT JOB Command Line Syntax

Figure 4.4, below, illustrates the command line syntax – using the command line, long form of the configuration options – of the SUBMIT JOB command.

```
-sub {filename | -jobname jobname -jobid jobid}
    [-target_jobname jobname]
    [-start
        [-immediate]
        [-targetserver server]
        [-target_variant job step,variant name;job step,variant name;...]
        [-wait
            [-poll seconds]
            [-joblog {yes|no}]
            [-spoollist {yes|no}]
            [-purge]
            [-waitchild {yes|no}]
            [-max_child_depth depth]
            [-joblogchild {yes|no|error}]
            [-spoollistchild {yes|no}]
            [-purgechild {yes|no}]
        ]
    ]
```

Figure 4.4 SUBMIT JOB Command Line Syntax

SUBMIT JOB Command Argument

The SUBMIT JOB command can be expressed as either:

- -U Short form
- -sub Long form

The SUBMIT JOB command argument, *filename*, specifies the name of the file that contains the job definition.

See Chapter 5 Job Definition Files for additional information on the job definition file.

SUBMIT JOB Command Options

Option Name	Description
IMMEDIATE_JOB	Causes the job to be started immediately.
JOB_ID	Job ID of an existing SAP job to use as a model for the new job definition.
JOB_LOG_CHILD	Controls the printing of job logs for child jobs.
JOB_NAME	Name of an existing SAP job to use as a model for the new job definition.
MAX_CHILD_DEPTH	Controls the maximum relationship depth that will be monitored by USAP.
PURGE_CHILD_JOBS	Controls the purging of child jobs.
PURGE_JOB	Purge job that has completed processing from SAP system
RETURN_JOB_LOG	Specification for whether or not the job's joblog is returned.
RETURN_SPOOL_LIST	Specification for whether or not the spoollists of all job steps are returned.
SPOOL_LIST_CHILD	Controls the printing of spoollists for child jobs.
START_JOB	Starts the newly defined job.
TARGET_JOB_NAME	Name to give the newly created job.
TARGET_VARIANT	One or more replacement variants for ABAP program job steps in an SAP job.
WAIT	Wait for the SAP job to complete processing.
WAIT_FOR_CHILD_JOBS	Controls the monitoring of child jobs.

Table 4.4 SUBMIT JOB Command Options

4.3.5 SUBMIT VARIANT Command

The SUBMIT VARIANT command defines a new variant to an SAP system for a specified ABAP report.

Note: SUBMIT VARIANT requires XBP interface 2.0.

(See Section 2.5 Client Fault Tolerance for information on XBP interface 2.0.)

SUBMIT VARIANT Command Line Syntax

Figure 4.5, below, illustrates the command line syntax – using the command line, long form of the configuration options – of the SUBMIT VARIANT command.

```
-sub filename
```

Figure 4.5 SUBMIT VARIANT Command Line Syntax

SUBMIT VARIANT Command Argument

The SUBMIT VARIANT command can be expressed as either:

- -U Short form
- -sub Long form

The SUBMIT VARIANT command argument, *filename*, specifies the name of the file that contains the variant definition.

See Section 5.10 Variant Definition File for additional information on the variant definition file.

4.3.6 SUBMIT INTERCEPT CRITERIA TABLE Command

The SUBMIT INTERCEPT CRITERIA TABLE command appends or replaces the SAP intercept criteria table.

SUBMIT INTERCEPT CRITERIA TABLE Command Line Syntax

Figure 4.6, below, illustrates the command line syntax – using the command line, long form of the configuration options – of the SUBMIT INTERCEPT CRITERIA TABLE command.

```
-sub filename
```

Figure 4.6 SUBMIT INTERCEPT CRITERIA TABLE Command Line Syntax

SUBMIT INTERCEPT CRITERIA TABLE Command Argument

The SUBMIT INTERCEPT CRITERIA TABLE command can be expressed as either:

- -U Short form
- -sub Long form

The SUBMIT INTERCEPT CRITERIA TABLE command argument, *filename*, specifies the name of the file that contains the intercept criteria table definition.

See Section 5.11 Job Intercept Table Definition File for additional information on the variant definition file.

4.3.7 SUBMIT FS JOBNET Command

The SUBMIT FS JOBNET command defines a new FS jobnet to an SAP system.

SUBMIT FS JOBNET Command Line Syntax

Figure 4.7, below, illustrates the command line syntax – using the command line, long form of the configuration options – of the SUBMIT FS JOBNET command.

```
-sub {filename | -jobname jobname -jobid jobid}
    [-start
        [-wait
             [-poll seconds]
             [-purge]
        ]
    ]
```

Figure 4.7 SUBMIT FS JOBNET Command Line Syntax

SUBMIT FS JOBNET Command Argument

The SUBMIT FS JOBNET command can be expressed as either:

- -U Short form
- -sub Long form

The SUBMIT FS JOBNET command argument, *filename*, specifies the name of the file that contains the FS jobnet definition.

See Section 5.12 FS Job Network Definition File for additional information on the variant definition file.

SUBMIT FS JOBNET Command Options

Option Name	Description
PURGE_JOB	Purge job that has completed processing from SAP system.
START_JOB	Starts the newly defined job.
USAP_POLL	Length of time to wait between job status calls to the SAP system.
WAIT	Wait for the SAP job to complete processing.

Table 4.5 SUBMIT FS JOBNET Command Options

4.3.8 MODIFY JOB Command

The MODIFY JOB command is used to modify an SAP job that already exists in an SAP system. A USAP job definition file is used to specify the modifications.

Job definition files are used to define new SAP jobs and to modify existing SAP jobs. The same syntactical rules apply to the job definition file in both cases with the following exceptions when modifying jobs:

1. SAP job identifier must be specified in order to identify the existing job to modify. The job identifier is specified in the job definition file using the JOBCOUNT keyword of the Job Header statement or the **`-jobid`** option of the MODIFY command. If both are used, the **`-jobid`** option overrides the JOBCOUNT value.
2. ABAP Step and External Step job definition statements must specify the step number of the existing job step to modify. The step number is specified using the STEP_NUMBER keyword of the ABAP Step and External Step job definition statements.

The parameter values specified in job definition file replace existing values in the SAP job definition. If a parameter is not specified in the job definition file, no change is made to the corresponding value in the existing SAP job definition.

See Chapter 5 Job Definition Files for additional information on the job definition file.

MODIFY JOB Command Line Syntax

Figure 4.8, below, illustrates the command line syntax – using the command line, long form of the configuration options – of the MODIFY JOB command.

```
-modify filename
    [-jobid jobid]
    [-start
        [-immediate]
        [-targetserver server]
        [-wait
            [-poll seconds]
            [-joblog {yes|no}]
            [-spoollist {yes|no}]
            [-purge]
            [-waitchild {yes|no}]
            [-joblogchild {yes|no|error}]
            [-spoollistchild {yes|no}]
            [-purgechild {yes|no}]
        ]
    ]
```

Figure 4.8 MODIFY JOB Command Line Syntax

MODIFY JOB Command Argument

The MODIFY JOB command can be expressed as either:

- -M Short form
- -modify Long form

The MODIFY JOB command argument, *filename*, specifies the name of the job definition file that contains the modification information.

See Chapter 5 Job Definition Files for additional information on the variant definition file.

MODIFY JOB Command Options

Option Name	Description
IMMEDIATE_JOB	Causes the job to be started immediately.
JOB_ID	Job ID of an existing SAP job to be modified.
JOB_LOG_CHILD	Controls the printing of job logs for child jobs.
PURGE_CHILD_JOBS	Controls the purging of child jobs.
PURGE_JOB	Purge job that has completed processing from SAP system.
RETURN_JOB_LOG	Specification for whether or not the job's joblog is returned.
RETURN_SPOOL_LIST	Specification for whether or not the spoollists of all job steps are returned.
SPOOL_LIST_CHILD	Controls the printing of spoollists for child jobs.
START_JOB	Starts the newly defined job.
TARGET_SERVER	Server on which the job will run.
USAP_POLL	Length of time to wait between job status calls to the SAP system.
WAIT	Wait for the SAP job to complete processing.
WAIT_FOR_CHILD_JOBS	Controls the monitoring of child jobs.

Table 4.6 MODIFY JOB Command Options

4.3.9 MODIFY VARIANT Command

The MODIFY VARIANT command is used to modify an SAP variant that already exists in an SAP system. A USAP variant definition file is used to specify the modifications.

Note: MODIFY VARIANT requires XBP interface 2.0.

(See Section 2.5 Client Fault Tolerance for information on XBP interface 2.0.)

Variant definition files are used to define new SAP variants and to modify existing SAP variants. The same syntactical rules apply to the variant definition file in both cases.

The parameter values specified in a variant definition file replace existing values in the SAP variant definition. If a parameter is not specified in the variant definition file, no change is made to the corresponding value in the existing SAP variant definition.

See Section 5.10 Variant Definition File for additional information on the variant definition file.

MODIFY VARIANT Command Line Syntax

Figure 4.9, below, illustrates the command line syntax – using the command line, long form of the configuration options – of the MODIFY VARIANT command.

```
-modify filename
```

Figure 4.9 MODIFY VARIANT Command Line Syntax

MODIFY VARIANT Command Argument

The MODIFY VARIANT command can be expressed as either:

- -M Short form
- -modify Long form

The MODIFY VARIANT command argument, *filename*, specifies the name of the variant definition file that contains the modification information.

See Section 5.10 Variant Definition File for additional information on the variant definition file.

4.3.10 START JOB Command

The START JOB command starts a currently defined SAP job.

START JOB Command Line Syntax

Figure 4.10, below, illustrates the command line syntax – using the command line, long form of the configuration options – of the START JOB command.

```
-start -jobname jobname -jobid jobid
    [-immediate]
    [-targetserver server]
    [-wait
        [-poll seconds]
        [-joblog {yes|no}]
        [-spoollist {yes|no}]
        [-purge]
        [-terminatedec exitcode]
        [-finishedec exitcode]
    ]
    [-bdcwait
         [-bdcjobnameptrn pattern]
         [-bdcjobidptrn pattern]
         [-bdcqidptrn pattern]
         [-qtobecreatedec exitcode]
         [-qunprocessedec exitcode]
         [-qinbackgroundec exitcode]
         [-qfinishedec exitcode]
         [-qerrorec exitcode]
    ]
```

Figure 4.10 START JOB Command Line Syntax

START JOB Command Options

Option Name	Description
BATCH_MONITOR	Causes USAP to perform batch input monitoring for the job being started.
EXIT_JOB_FINISHED	USAP exit code for the SAP job finished status.
EXIT_JOB_TERMINATED	USAP exit code for the SAP job terminated status.
EXIT_QUEUE_BACKGROUND	USAP exit code for the SAP queue state ‘S’ (in background)
EXIT_QUEUE_CREATED	USAP exit code for the SAP queue state ‘C’ (to be created).
EXIT_QUEUE_ERROR	USAP exit code for the SAP queue state ‘E’ (error).
EXIT_QUEUE_FINISHED	USAP exit code for the SAP queue state ‘F’ (finished).
EXIT_QUEUE_UNPROCESSED	USAP exit code for the SAP queue state ‘ ’ (unprocessed).
IMMEDIATE_JOB	Causes the job to be started immediately.
JOB_ID	Job ID of an existing SAP job to use as a model for the new job definition.
JOB_LOG_CHILD	Controls the printing of job logs for child jobs.
JOB_NAME	Name of an existing SAP job to use as a model for the new job definition.
JOB_ID_PATTERN	Locates the header record and determines the offset of the job ID in the RSBDCSUB batch input processing report.
JOB_NAME_PATTERN	Locates the header record and determines the offset of the job name in the RSBDCSUB batch input processing report.
PURGE_CHILD_JOBS	Controls the purging of child jobs.
PURGE_JOB	Purge job that has completed processing from SAP system.
QUEUE_ID_PATTERN	Locates the header record and determines the offset of the queue ID in the RSBDCSUB batch input processing report.
RETURN_JOB_LOG	Specification for whether or not the job’s joblog is returned.
RETURN_SPOOL_LIST	Specification for whether or not the spoollists of all job steps are returned.
SPOOL_LIST_CHILD	Controls the printing of spoollists for child jobs.
TARGET_SERVER	Server on which the job will run.
USAP_POLL	Length of time to wait between job status calls to the SAP system.
WAIT	Wait for the SAP job to complete processing.
WAIT_FOR_CHILD_JOBS	Controls the monitoring of child jobs.

Table 4.7 START JOB Command Options

4.3.11 START FS JOBNET Command

The START FS JOBNET command starts a specified FS job network on an SAP system.

START FS JOBNET Command Line Syntax

Figure 4.11, below, illustrates the command line syntax – using the command line, long form of the configuration options – of the START FS JOBNET command.

```
-start -jnetid jobnet_id -jnetprcid jobnet_process_id
    [-wait
        [-poll seconds]
        [-purge]
        ]
```

Figure 4.11 START FS JOBNET Command Line Syntax

START FS JOBNET Command Options

Option Name	Description
JOB_NETWORK_ID	Network identifier for the pre-existing SAP FS job network being started.
JOB_PROCESS_ID	Process ID of an existing SAP FS job network process to start.
PURGE_JOB	Purge job that has completed processing from SAP system.
USAP_POLL	Length of time to wait between job status calls to the SAP system.
WAIT	Wait for the SAP job to complete processing.

Table 4.8 START FS JOBNET Command Options

4.3.12 WAIT for JOB Command

The WAIT for JOB command allows USAP to reconnect to a started job and monitor it through completion.

WAIT for JOB Command Line Syntax

Figure 4.12, below, illustrates the command line syntax – using the command line, long form of the configuration options – of the WAIT for JOB command.

```
-wait -jobname jobname -jobid jobid
    [-job_stat_check_interval seconds]
    [-joblog {yes|no}]
    [-applog {yes|no}]
    [-printapprc {yes|no}]
    [-useapprc {yes|no}]
    [-spoollist {yes|no}]
    [-spool_codepage codepage]
    [-transtab translation_table]
    [-terminatedec exitcode]
    [-finishedec exitcode]
    [-poll seconds]
    [-purge]
    [-syslog {yes|no}
        [-syslogpre seconds]
        [-syslogpost seconds]
    ]
    [-waitchild {yes|no}]
    [-max_child_depth depth]
    [-joblogchild {yes|no|error}]
    [-spoollistchild {yes|no}]
    [-purgechild {yes|no}]
    [-max_log_size size]
    [-max_spool_size size]
```

Figure 4.12 WAIT for JOB Command Line Syntax

WAIT for JOB Command Options

Option Name	Description
WAIT	Causes USAP to wait for the SAP job to complete processing
EXIT_JOB_FINISHED	USAP exit code for the SAP job finished status.
EXIT_JOB_TERMINATED	USAP exit code for the SAP job terminated status.
JOB_ID	Job ID of an existing SAP job to use as a model for the new job definition.
JOB_LOG_CHILD	Controls the printing of job logs for child jobs.
JOB_NAME	Name of an existing SAP job to use as a model for the new job definition.
MAX_CHILD_DEPTH	Controls the maximum relationship depth that will be monitored by USAP.
MAX_JOB_LOG_SIZE	Maximum size for job logs.
MAX_SPOOL_LIST_SIZE	Maximum size for spool lists.
PURGE_CHILD_JOBS	Controls the purging of child jobs.
PURGE_JOB	Purge job that has completed processing from SAP system.
RETURN_APPLICATION_LOG	Specification for whether or not the job's application log is returned.
RETURN_APPLICATION_RC	Specification for whether or not the job's application return codes are returned.
RETURN_JOB_LOG	Specification for whether or not the job's joblog is returned.
RETURN_SPOOL_LIST	Specification for whether or not the spoollists of all job steps are returned.
SPOOL_CODEPAGE	Codepage used for transferring spool lists from SAP system.
SPOOL_LIST_CHILD	Controls the printing of spoollists for child jobs.
STATUS_CHECK_INTERVAL	Length of time that can elapse, without a change in job status, before a call will be made to synchronize the actual job status with the SAP stored status.
SYSLOG	Specification for whether or not a syslog report is generated on standard error if the job does not complete successfully.
SYSLOG_POST_TIME	Length of time to add to the job **`end`** time when calculating the **`to`** time for the syslog report.
SYSLOG_PRE_TIME	Length of time to subtract from the job release time when calculating the **`from`** time for the syslog report.
TRANSLATION_TABLE	Spoollist translation table file to use for formatting returned spoollists.
USAP_POLL	Length of time to wait between job status calls to the SAP system.
USE_APPLICATION_RC	Specification for whether or not the job's application return codes are used to determine the exit code of the USAP job.
WAIT_FOR_CHILD_JOBS	Controls the monitoring of child jobs.

Table 4.9 WAIT for JOB Command Options

4.3.13 MASS ACTIVITY WAIT Command

The MASS ACTIVITY WAIT command allows USAP to wait for (or reconnect and wait for) a started mass activity job and monitor it, and all its interval jobs, through completiont.

MASS ACTIVITY WAIT Command Line Syntax

Figure 4.13, below, illustrates the command line syntax – using the command line, long form of the configuration options – of the MASS ACTIVITY WAIT command.

```
-mawait -jobname jobname -jobid jobid
    [-poll seconds]
    [-job_stat_check_interval seconds]
    [-joblog {yes|no}]
    [-applog {yes|no}]
    [-printapprc {yes|no}]
    [-useapprc {yes|no}]
    [-transtab translation_table]
    [-purge]
    [-syslog {yes|no}
        [-syslogpre seconds]
        [-syslogpost seconds]
    ]
    [-terminatedec exitcode]
    [-finishedec exitcode]
```

Figure 4.13 MASS ACTIVITY WAIT Command Line Syntax

MASS ACTIVITY WAIT Command Options

Option Name	Description
MASS_ACTIVITY_WAIT	Causes USAP to wait for the SAP mass activity jobs to complete processing.
EXIT_JOB_FINISHED	USAP exit code for the SAP job finished status.
EXIT_JOB_TERMINATED	USAP exit code for the SAP job terminated status.
JOB_ID	Job ID of an existing SAP job to use as a model for the new job definition.
JOB_NAME	Name of an existing SAP job to use as a model for the new job definition.
PURGE_JOB	Purge job that has completed processing from SAP system.
RETURN_APPLICATION_LOG	Specification for whether or not the job's application log is returned.
RETURN_APPLICATION_RC	Specification for whether or not the job's application return codes are returned.
RETURN_JOB_LOG	Specification for whether or not the job's joblog is returned.
STATUS_CHECK_INTERVAL	Length of time that can elapse, without a change in job status, before a call will be made to synchronize the actual job status with the SAP stored status.
SYSLOG	Specification for whether or not a syslog report is generated on standard error if the job does not complete successfully.
SYSLOG_POST_TIME	Length of time to add to the job **end** time when calculating the **to** time for the syslog report.
SYSLOG_PRE_TIME	Length of time to subtract from the job release time when calculating the **from** time for the syslog report.
TRANSLATION_TABLE	Spoollist translation table file to use for formatting returned spoollists.
USAP_POLL	Length of time to wait between job status calls to the SAP system.
USE_APPLICATION_RC	Specification for whether or not the job's application return codes are used to determine the exit code of the USAP job.

Table 4.10 MASS ACTIVITY WAIT Command Options

4.3.14 WAIT for FS JOB NETWORK Command

The WAIT for FS JOB NETWORK command allows USAP to reconnect to a started FS job network and monitor it through completion.

WAIT for FS JOB NETWORK Command Line Syntax

Figure 4.14, below, illustrates the command line syntax – using the command line, long form of the configuration options – of the WAIT for FS JOB NETWORK command.

```
-wait -jnetid jobnetid -jnetprcid processid
      [-poll seconds]
    [-purge]
    [-syslog {yes|no}
        [-syslogpre seconds]
        [-syslogpost seconds]
    ]
    [-max_log_size size]
    [-max_spool_size size]
```

Figure 4.14 WAIT for FS JOB NETWORK Command Line Syntax

WAIT for FS JOB NETWORK Command Options

Option Name	Description
WAIT	Causes USAP to wait for the SAP job network to complete processing.
JOB_NETWORK_ID	Network identifier for the pre-existing SAP FS job network being started.
JOB_PROCESS_ID	Process ID of an existing SAP FS job network process to start.
MAX_JOB_LOG_SIZE	Maximum size for job logs.
MAX_SPOOL_LIST_SIZE	Maximum size for spool lists.
PURGE_JOB	Purge job that has completed processing from SAP system.
SYSLOG	Specification for whether or not a syslog report is generated on standard error if the job does not complete successfully.
SYSLOG_POST_TIME	Length of time to add to the job end time when calculating the **to** time for the syslog report.
SYSLOG_PRE_TIME	Length of time to subtract from the job release time when calculating the **from** time for the syslog report.
USAP_POLL	Length of time to wait between job status calls to the SAP system.

Table 4.11 WAIT for FS JOB NETWORK Command Options

4.3.15 BDCWAIT Command

The BDCWAIT command allows USAP to reconnect to a started batch input processing job and monitor it, and all its generated session processing jobs, through completion.

BDCWAIT Command Line Syntax

Figure 4.15, below, illustrates the command line syntax – using the command line, long form of the configuration options – of the BDCWAIT command.

```
-bdcwait -jobname jobname -jobid jobid
    [-poll seconds]
    [-job_stat_check_interval seconds]
    [-joblog {yes|no}]
    [-applog {yes|no}]
    [-printapprc {yes|no}]
    [-useapprc {yes|no}]
    [-transtab translation_table]
    [-purge]
    [-purge_bdc_map {yes|no}]
    [-syslog {yes|no}
        [-syslogpre seconds]
        [-syslogpost seconds]
    ]
    [-terminatedec exitcode]
    [-finishedec exitcode]
    [-qtobecreatedec exitcode]
    [-qunprocessedec exitcode]
    [-qinbackgroundec exitcode]
    [-qfinishedec exitcode]
    [-qerrorec exitcode]
    [-bdcjobnameptrn pattern]
    [-bdcjobidptrn pattern]
    [-bdcqidptrn pattern]
```

Figure 4.15 BDCWAIT Command Line Syntax

BDCWAIT Command Options

Option Name	Description
BATCH_MONITOR	Causes USAP to perform batch input monitoring for the job being started.
EXIT_JOB_FINISHED	USAP exit code for the SAP job finished status.
EXIT_JOB_TERMINATED	USAP exit code for the SAP job terminated status.
EXIT_QUEUE_BACKGROUND	USAP exit code for the SAP queue state ‘S’ (in background).
EXIT_QUEUE_CREATED	USAP exit code for the SAP queue state ‘C’ (to be created).
EXIT_QUEUE_ERROR	USAP exit code for the SAP queue state ‘E’ (error).
EXIT_QUEUE_FINISHED	USAP exit code for the SAP queue state ‘F’ (finished).
EXIT_QUEUE_UNPROCESSED	USAP exit code for the SAP queue state ‘ ’ (unprocessed).
JOB_ID	Job ID of an existing SAP job to use as a model for the new job definition.
JOB_NAME	Name of an existing SAP job to use as a model for the new job definition.
JOB_ID_PATTERN	Locates the header record and determines the offset of the job ID in the **RSBDCSUB** batch input processing report.
JOB_NAME_PATTERN	Locates the header record and determines the offset of the job name in the **RSBDCSUB** batch input processing report.
PURGE_BDC_MAP	Specification for whether or not to delete BDC Batch input session queues that have been processed successfully.
PURGE_JOB	Purge job that has completed processing from SAP system.
QUEUE_ID_PATTERN	Locates the header record and determines the offset of the queue ID in the **RSBDCSUB** batch input processing report.
RETURN_APPLICATION_LOG	Specification for whether or not the job’s application log is returned.
RETURN_APPLICATION_RC	Specification for whether or not the job’s application return codes are returned.
RETURN_JOB_LOG	Specification for whether or not the job’s joblog is returned.
STATUS_CHECK_INTERVAL	Length of time that can elapse, without a change in job status, before a call will be made to synchronize the actual job status with the SAP stored status.
SYSLOG	Specification for whether or not a syslog report is generated on standard error if the job does not complete successfully.
SYSLOG_POST_TIME	Length of time to add to the job end time when calculating the **to** time for the syslog report.
SYSLOG_PRE_TIME	Length of time to subtract from the job release time when calculating the **from** time for the syslog report.
TRANSLATION_TABLE	Spoollist translation table file to use for formatting returned spoollists.
USAP_POLL	Length of time to wait between job status calls to the SAP system.
USE_APPLICATION_RC	Specification for whether or not the job’s application return codes are used to determine the exit code of the USAP job.

Table 4.12 BDCWAIT Command Options

4.3.16 ABORT Command

The ABORT command cancels a running SAP job.

ABORT Command Line Syntax

Figure 4.16, below, illustrates the command line syntax – using the command line, long form of the configuration options – of the ABORT command.

```
-abort -jobname jobname -jobid jobid
```

Figure 4.16 ABORT Command Line Syntax

ABORT Command Options

Option Name	Description
JOB_ID	Job ID of an existing SAP job to use as a model for the new job definition.
JOB_NAME	Name of an existing SAP job to use as a model for the new job definition.

Table 4.13 ABORT Command Options

4.3.17 PURGE JOB Command

The PURGE JOB command deletes a defined SAP job, its joblog, and all of its spoollists.

This command is not available on SAP 3.1 and SAP 4.0 systems.

PURGE JOB Command Line Syntax

Figure 4.17, below, illustrates the command line syntax – using the command line, long form of the configuration options – of the PURGE JOB command.

```
-purge -jobname jobname -jobid jobid
```

Figure 4.17 PURGE JOB Command Line Syntax

PURGE JOB Command Options

Option Name	Description
JOB_ID	Job ID of an existing SAP job to use as a model for the new job definition.
JOB_NAME	Name of an existing SAP job to use as a model for the new job definition.

Table 4.14 PURGE JOB Command Options

4.3.18 PURGE FS JOB NETWORK Command

The PURGE FS JOB NETWORK command deletes a defined SAP FS job network.

PURGE FS JOB NETWORK Command Line Syntax

Figure 4.18, below, illustrates the command line syntax – using the command line, long form of the configuration options – of the PURGE FS JOB NETWORK command.

```
-purge -jnetid jobnetid -jnetprcid processid
```

Figure 4.18 PURGE FS JOB NETWORK Command Line Syntax

PURGE FS JOB NETWORK Command Options

Option Name	Description
JOB_NETWORK_ID	Network identifier for the pre-existing SAP FS job network being started.
JOB_PROCESS_ID	Process ID of an existing SAP FS job network process to start.

Table 4.15 PURGE FS JOB NETWORK Command Options

4.3.19 RAISE EVENT Command

The RAISE EVENT command raises the specified SAP background processing event.

RAISE EVENT Command Line Syntax

Figure 4.19, below, illustrates the command line syntax – using the command line, long form of the configuration options – of the RAISE EVENT command.

```
-raise_bp_event –event_id id –event_parm parm
```

Figure 4.19 RAISE EVENT Command Line Syntax

RAISE EVENT Command Options

Option Name	Description
EVENT_ID	Name of the event.
EVENT_PARAMETER	Optional parameter value for the event.

Table 4.16 RAISE EVENT Command Options

4.3.20 SYNTAX Command

The SYNTAX command checks the syntax of a USAP definition file.

SYNTAX Command Line Syntax

Figure 4.20, below, illustrates the command line syntax – using the command line, long form of the configuration options – of the SYNTAX command.

```
-syntax filename
```

Figure 4.20 SYNTAX Command Line Syntax

SYNTAX Command Argument

The SYNTAX command can be expressed as either:

- -X Short form
- -syntax Long form

The SYNTAX command argument, *filename*, specifies the name of the definition file that contains the job, variant, or FS job network definition.

- See Chapter 5 Job Definition Files for additional information on the job definition file.
- See Section 5.10 Variant Definition File for additional information on the variant definition file.
- See Section 5.12 FS Job Network Definition File for additional information on the FS Job Network definition file.

4.3.21 DISPLAY JOBLOG Command

The DISPLAY JOBLOG command displays the job log for a specified SAP job.

DISPLAY JOBLOG Command Line Syntax

Figure 4.21, below, illustrates the command line syntax – using the command line, long form of the configuration options – of the DISPLAY JOBLOG command.

```
-display joblog -jobname jobname -jobid jobid -max_log_size size
```

Figure 4.21 DISPLAY JOBLOG Command Line Syntax

DISPLAY JOBLOG Command Argument

The DISPLAY JOBLOG command can be expressed as either:

- -D Short form
- -display Long form

The DISPLAY JOBLOG command argument, **joblog**, requests the display of a job's joblog.

DISPLAY JOBLOG Command Options

Option Name	Description
JOB_ID	Job ID of an existing SAP job to use as a model for the new job definition.
JOB_NAME	Name of an existing SAP job to use as a model for the new job definition.
MAX_JOB_LOG_SIZE	Maximum size for job logs.

Table 4.17 DISPLAY JOBLOG Command Options

4.3.22 DISPLAY SPOOLLIST Command

The DISPLAY SPOOLLIST command displays the spoollist for a job step.

DISPLAY SPOOLLIST Command Line Syntax

Figure 4.22, below, illustrates the command line syntax – using the command line, long form of the configuration options – of the DISPLAY SPOOLLIST command.

```
-display spoollist –jobname jobname –jobid jobid
         –stepnum stepnumber -max_spool_size size
         [-spool_codepage codepage]
         [–transtab translation_table]
```

Figure 4.22 DISPLAY SPOOLLIST Command Line Syntax

DISPLAY SPOOLLIST Command Argument

The DISPLAY SPOOLLIST command can be expressed as either:

- -D Short form
- -display Long form

The DISPLAY SPOOLLIST command argument, **spoollist**, requests the display of a job step's spoollist.

DISPLAY SPOOLLIST Command Options

Option Name	Description
JOB_ID	Job ID of an existing SAP job to use as a model for the new job definition.
JOB_NAME	Name of an existing SAP job to use as a model for the new job definition.
MAX_SPOOL_LIST_SIZE	Maximum size for spool lists.
SPOOL_CODEPAGE	Codepage used for transferring spool lists from SAP system.
STEP_NUMBER	Step number of the SAP job step.
TRANSLATION_TABLE	spoollist translation table file to use for formatting returned spoollists.

Table 4.18 DISPLAY SPOOLLIST Command Options

4.3.23 DISPLAY STATUS Command

The DISPLAY STATUS command displays the current status for an SAP job. The status is printed to standard output and the exit code of **usap** indicates the status. See Section 4.5 Exit Codes for a complete list of job status exit codes.

DISPLAY STATUS Command Line Syntax

Figure 4.23, below, illustrates the command line syntax – using the command line, long form of the configuration options – of the DISPLAY STATUS command.

```
-display status -jobname jobname -jobid jobid
    [-activeec exitcode     ]
    [-readyec exitcode      ]
    [-scheduledec exitcode  ]
    [-releasedec exitcode   ]
    [-terminatedec exitcode ]
    [-finishedec exitcode   ]
```

Figure 4.23 DISPLAY STATUS Command Line Syntax

DISPLAY STATUS Command Argument

The DISPLAY STATUS command can be expressed as either:

- -D Short form
- -display Long form

The DISPLAY STATUS command argument, **status**, requests a job status.

The status is printed to standard output and the exit code of USAP indicates the status. See Section 4.5 Exit Codes for a complete list of job status exit codes.

DISPLAY STATUS Command Options

Option Name	Description
EXIT_JOB_ACTIVE	USAP exit code for the SAP job **`active`** status.
EXIT_JOB_FINISHED	USAP exit code for the SAP job **`finished`** status.
EXIT_JOB_READY	USAP exit code for the SAP job **`ready`** status.
EXIT_JOB_RELEASED	USAP exit code for the SAP job **`released`** status.
EXIT_JOB_SCHEDULED	USAP exit code for the SAP job **`scheduled`** status.
EXIT_JOB_TERMINATED	USAP exit code for the SAP job **`terminated`** status.
JOB_ID	Job ID of an existing SAP job to use as a model for the new job definition.
JOB_NAME	Name of an existing SAP job to use as a model for the new job definition.

Table 4.19 DISPLAY STATUS Command Options

4.3.24 DISPLAY VARIANTS Command

The DISPLAY VARIANTS command displays the variants available for the specified ABAP program.

DISPLAY VARIANTS Command Line Syntax

Figure 4.24, below, illustrates the command line syntax – using the command line, long form of the configuration options – of the DISPLAY VARIANTS command.

```
-display variants -abapname abapname -varselopt {A|B}
```

Figure 4.24 DISPLAY VARIANTS Command Line Syntax

DISPLAY VARIANTS Command Argument

The DISPLAY VARIANTS command can be expressed as either:

- -D Short form
- -display Long form

The DISPLAY VARIANTS command argument, **variants**, displays the variants defined for ABAP program **abapname**.

- Using **-varselopt** *A* will display the variants that are available for batch and dialog mode.
- Using **-varselopt** *B* will display the variants that are available for batch mode only.

DISPLAY VARIANTS Command Options

Option Name	Description
ABAP_NAME	Name of an ABAP program in an SAP system.
VARIANT_SELECTION	Specification to display either variants available for batch and dialog mode or variants available only for batch mode.

Table 4.20 DISPLAY VARIANTS Command Options

4.3.25 DISPLAY VARIANT Command

The DISPLAY VARIANT command displays the contents of a specified variant.

Note: DISPLAY VARIANT requires XBP interface 2.0.
(See Section 2.5 Client Fault Tolerance for information on XBP interface 2.0.)

DISPLAY VARIANT Command Line Syntax

Figure 4.25, below, illustrates the command line syntax – using the command line, long form of the configuration options – of the DISPLAY VARIANT command.

```
-display variant –variant variantname -varlang language
–abapname abapname
```

Figure 4.25 DISPLAY VARIANT Command Line Syntax

DISPLAY VARIANT Command Argument

The DISPLAY VARIANT command can be expressed as either:

- -D Short form
- -display Long form

The DISPLAY VARIANT command argument, **variant**, displays the specified SAP variant.

DISPLAY VARIANT Command Options

Option Name	Description
ABAP_NAME	Name of an ABAP program in an SAP system.
VARIANT	Pre-existing SAP variant whose contents will be displayed.
VARIANT_LANGUAGE	Preferred language in which to return the variant description.

Table 4.21 DISPLAY VARIANT Command Options

4.3.26 DISPLAY JOBDEF Command

The DISPLAY JOBDEF command displays the definition of the specified SAP job.

DISPLAY JOBDEF Command Line Syntax

Figure 4.26, below, illustrates the command line syntax – using the command line, long form of the configuration options – of the DISPLAY JOBDEF command.

```
-display jobdef -jobname jobname -jobid jobid
```

Figure 4.26 DISPLAY JOBDEF Command Line Syntax

DISPLAY JOBDEF Command Argument

The DISPLAY JOBDEF command can be expressed as either:

- -D Short form
- -display Long form

The DISPLAY JOBDEF command argument, **jobdef**, requests the display of a job's definition.

DISPLAY JOBDEF Command Options

Option Name	Description
JOB_ID	Job ID of an existing SAP job to use as a model for the new job definition.
JOB_NAME	Name of an existing SAP job to use as a model for the new job definition.

Table 4.22 DISPLAY JOBDEF Command Options

4.3.27 DISPLAY SELECT Command

The DISPLAY SELECT command displays a variety of attributes for a list of SAP jobs that match the specified criteria.

DISPLAY SELECT Command Line Syntax

Figure 4.27, below, illustrates the command line syntax – using the command line, long form of the configuration options – of the DISPLAY SELECT command.

```
-display select -jobname jobmask
    [-jobid idmask             ]
    [-selusername userid       ]
    [-fromdate date            ]
    [-todate date              ]
    [-fromtime time            ]
    [-totime time              ]
    [-nodate {yes|no}          ]
    [-withpred {yes|no}        ]
    [-released {yes|no}        ]
    [-scheduled {yes|no}       ]
    [-ready {yes|no}           ]
    [-running {yes|no}         ]
    [-finished {yes|no}        ]
    [-aborted {yes|no}         ]
    [-output output-field-list]
```

Figure 4.27 DISPLAY SELECT Command Line Syntax

DISPLAY SELECT Command Argument

The DISPLAY SELECT command can be expressed as either:

- -D Short form
- -display Long form

The DISPLAY SELECT command argument, **select**, requests the display of all jobs matching the **jobmask** and any additional selection criteria specified. The default output for this command is the job name and job ID for each job found. However, additional fields can be printed using the **–output** option.

Note: This command is not available on SAP 3.1 and SAP 4.0 systems.

DISPLAY SELECT Command Options

Option Name	Description
FROM_DATE	Earliest date to use for job selection or syslog request.
FROM_TIME	Earliest time to use for job selection or syslog request.
JOB_ID	Job ID of an existing SAP job to use as a model for the new job definition.
JOB_NAME	Name of an existing SAP job to use as a model for the new job definition.
NO_START_DATE	Specification for whether or not to include jobs with no start date in selection criteria.
OUTPUT_FIELD_LIST	Additional fields to write for the **`select`** command.
STATUS_ABORTED	Specification for whether or not to include jobs with status aborted in selection criteria.
STATUS_FINISHED	Specification for whether or not to include jobs with status finished in selection criteria.
STATUS_READY	Specification for whether or not to include jobs with status ready in selection criteria.
STATUS_RELEASED	Specification for whether or not to include jobs with status released in selection criteria.
STATUS_RUNNING	Specification for whether or not to include jobs with status running in selection criteria.
STATUS_SCHEDULED	Specification for whether or not to include jobs with status scheduled in selection criteria.
TO_DATE	Latest date to use for job selection or syslog request.
TO_TIME	Latest time to use for job selection or syslog request.
USER_NAME	User ID associated with a job.
WITH_PREDECESSOR	Specification for whether or not to include jobs with start after predecessor in selection criteria.

Table 4.23 DISPLAY SELECT Command Options

4.3.28 DISPLAY SYSLOG Command

The DISPLAY SYSLOG command displays a portion of an SAP syslog that meets the specified date/time constraints.

DISPLAY SYSLOG Command Line Syntax

Figure 4.28, below, illustrates the command line syntax – using the command line, long form of the configuration options – of the DISPLAY SYSLOG command.

```
-display syslog -fromdate date -todate date
    [-fromtime time      ]
    [-totime time        ]
    [-pagelimit limit    ]
    [-targetserver server]
```

Figure 4.28 DISPLAY SYSLOG Command Line Syntax

DISPLAY SYSLOG Command Argument

The DISPLAY SYSLOG command can be expressed as either:

- -D Short form
- -display Long form

The DISPLAY SYSLOG command argument, **syslog**, requests entries from an SAP System syslog for a specified date and time range.

DISPLAY SYSLOG Command Options

Option Name	Description
FROM_DATE	Earliest date to use for job selection or syslog request.
FROM_TIME	Earliest time to use for job selection or syslog request.
PAGE_LIMIT	Maximum number of pages that can be returned in the syslog report.
TARGET_SERVER	Name of the server whose syslog will be read.
TO_DATE	Latest date to use for job selection or syslog request.
TO_TIME	Latest time to use for job selection or syslog request.

Table 4.24 DISPLAY SYSLOG Command Options

4.3.29 DISPLAY INTERCEPTED_JOBS Command

The DISPLAY INTERCEPTED_JOBS command displays the job intercept for the connected SAP system.

DISPLAY INTERCEPTED_JOBS Command Line Syntax

Figure 4.29, below, illustrates the command line syntax – using the command line, long form of the configuration options – of the DISPLAY INTERCEPTED_JOBS command.

```
-display intercepted_jobs -dspclient client
```

Figure 4.29 DISPLAY INTERCEPT_JOBS Command Line Syntax

DISPLAY INTERCEPTED_JOBS Command Argument

The DISPLAY INTERCEPTED_TABLE command can be expressed as either:

- -D Short form
- -display Long form

The DISPLAY INTERCEPTED_JOBS command argument, **intercepted_jobs**, requests the display of an SAP system's intercepted jobs. Unless a specific client is identified, intercepted jobs for all clients are displayed.

DISPLAY INTERCEPTED_JOBS Command Options

Option Name	Description
DISPLAY_CLIENT	Specific SAP client whose intercepted jobs will be reported.

Table 4.25 DISPLAY INTERCEPTED_JOBS Command Options

4.3.30 DISPLAY INTERCEPT_TABLE Command

The DISPLAY INTERCEPT_TABLE command displays the contents of the job intercept criteria table for the connected SAP system.

DISPLAY INTERCEPT_TABLE Command Line Syntax

Figure 4.30, below, illustrates the command line syntax – using the command line, long form of the configuration options – of the DISPLAY INTERCEPT_TABLE command.

```
-display intercept_table
```

Figure 4.30 DISPLAY INTERCEPT_TABLE Command Line Syntax

DISPLAY INTERCEPT_TABLE Command Argument

The DISPLAY INTERCEPT_TABLE command can be expressed as either:

- -D Short form
- -display Long form

The DISPLAY INTERCEPT_TABLE command argument, **intercept_table**, requests the display of an SAP system's job intercept criteria table.

4.3.31 DISPLAY QSTATE Command

The DISPLAY QSTATE command displays the state of a specific Batch Input / BDC session queue in an SAP system.

DISPLAY QSTATE Command Line Syntax

Figure 4.31, below, illustrates the command line syntax – using the command line, long form of the configuration options – of the DISPLAY QSTATE command.

```
-display qstate -qid queueid
```

Figure 4.31 DISPLAY QSTATE Command Line Syntax

DISPLAY QSTATE Command Argument

The DISPLAY QSTATE command can be expressed as either:

- -D Short form
- -display Long form

The DISPLAY QSTATE command argument, **qstate**, requests the state of a queue used to process a batch input session. See Section 4.5.4 DISPLAY QSTATE Exit Codes for a complete list of queue state exit codes.

DISPLAY QSTATE Command Options

Option Name	Description
QUEUE_ID	Queue identifier associated with the batch input session.

Table 4.26 DISPLAY QSTATE Command Options

4.3.32 DISPLAY REPORTS Command

The DISPLAY REPORTS command displays a list of ABAP reports that match the specified criteria.

DISPLAY REPORTS Command Line Syntax

Figure 4.32, below, illustrates the command line syntax – using the command line, long form of the configuration options – of the DISPLAY REPORTS command.

```
-display reports -abapname abapmask -count max_hit_count
```

Figure 4.32 DISPLAY REPORTS Command Line Syntax

DISPLAY REPORTS Command Argument

The DISPLAY REPORTS command can be expressed as either:

- -D Short form
- -display Long form

The DISPLAY REPORTS command argument, **reports**, requests the display of a list of ABAP reports that match the specified criteria.

DISPLAY REPORTS Command Options

Option Name	Description
ABAP_NAME	Complete ABAP name or a mask used to select SAP ABAP reports that match the mask.
MAX_HIT_COUNT	Maximum number of ABAP reports to be returned.

Table 4.27 DISPLAY REPORTS Command Options

4.3.33 DISPLAY COMMANDS Command

The DISPLAY COMMANDS command displays a list of SAP external commands that match the specified criteria.

DISPLAY COMMANDS Command Line Syntax

Figure 4.33, below, illustrates the command line syntax – using the command line, long form of the configuration options – of the DISPLAY COMMANDS command.

```
-display commands -cmd external_command_mask -opsys operating_system
```

Figure 4.33 DISPLAY COMMANDS Command Line Syntax

DISPLAY COMMANDS Command Argument

The DISPLAY COMMANDS command can be expressed as either:

- -D Short form
- -display Long form

The DISPLAY COMMANDS command argument, **commands**, requests the display of a list of SAP external commands that match the specified criteria.

DISPLAY COMMANDS Command Options

Option Name	Description
EXTERNAL_COMMAND	Complete command name or a mask used to select SAP external commands that match the mask.
OPERATING_SYSTEM	Name of the operating system for which external commands are searched.

Table 4.28 DISPLAY COMMANDS Command Options

4.3.34 DISPLAY OUTPUT_DEVICES Command

The DISPLAY OUTPUT_DEVICES command displays a list of SAP output devices that match the specified criteria.

DISPLAY OUTPUT_DEVICES Command Line Syntax

Figure 4.34, below, illustrates the command line syntax – using the command line, long form of the configuration options – of the DISPLAY OUTPUT_DEVICES command.

```
-display output_devices -short_name technical_device_name_mask
    -long_name long_device_name_mask
```

Figure 4.34 DISPLAY OUTPUT_DEVICES Command Line Syntax

DISPLAY OUTPUT_DEVICES Command Argument

The DISPLAY OUTPUT_DEVICES command can be expressed as either:

- -D Short form
- -display Long form

The DISPLAY OUTPUT_DEVICES command argument, **output_devices**, requests the display of a list of SAP output devices that match the specified criteria.

DISPLAY OUTPUT_DEVICES Command Options

Option Name	Description
TECHNICAL_DEVICE_NAME	Complete device name or a mask used to select SAP output devices that match the mask.
LONG_DEVICE_NAME	Complete device name or a mask used to select SAP output devices that match the mask.

Table 4.29 DISPLAY OUTPUT_DEVICES Command Options

4.3.35 DISPLAY PRINT_FORMATS Command

The DISPLAY PRINT_FORMATS command displays a list of print formats that are available for the specified printer.

DISPLAY PRINT_FORMATS Command Line Syntax

Figure 4.35, below, illustrates the command line syntax – using the command line, long form of the configuration options – of the DISPLAY PRINT_FORMATS command.

```
-display print_formats -printer printer_name -layout layout
```

Figure 4.35 DISPLAY PRINT_FORMATS Command Line Syntax

DISPLAY PRINT_FORMAT Command Argument

The DISPLAY PRINT_FORMAT command can be expressed as either:

- -D Short form
- -display Long form

The DISPLAY PRINT_FORMAT command argument, **print_formats**, requests the display of a list of print formats available for the specified printer.

DISPLAY PRINT_FORMATS Command Options

Option Name	Description
LAYOUT_NAME	Complete layout name or a mask used to select printer layouts that match the mask.
PRINTER_NAME	Name of a printer for which the print formats will be retrieved.

Table 4.30 DISPLAY PRINT_FORMATS Command Options

4.3.36 GENERATE JOB DEFINITION FILE Command

The GENERATE JOB DEFINITION FILE command generates a USAP job definition file based on a model SAP job. The generated definition file is written to standard output.

GENERATE JOB DEFINITION FILE Command Line Syntax

Figure 4.36, below, illustrates the command line syntax – using the command line, long form of the configuration options – of the GENERATE JOB DEFINITION FILE command.

```
-generate jobdef -jobname jobname -jobid jobid
```

Figure 4.36 GENERATE JOB DEFINITION FILE Command Line Syntax

GENERATE JOB DEFINITION FILE Command Argument

The GENERATE JOB DEFINITION FILE command can be expressed as:

- -generate Long form

The GENERATE JOB DEFINITION FILE command argument, **jobdef**, generates a usap job definition file based on the specified SAP job definition. The generated job definition is printed to standard output.

This command option makes it easy to create complex job definitions based on pre-existing SAP jobs. It also eliminates typing errors that can be introduced by manually coding job definition files.

GENERATE JOB DEFINITION FILE Command Options

Option Name	Description
JOB_NAME	Name of an existing SAP job to select as the model job.
JOB_ID	Job ID of an existing SAP job to select as the model job.

Table 4.31 GENERATE JOB DEFINITION FILE Command Options

4.3.37 GENERATE VARIANT DEFINITION FILE Command

The GENERATE VARIANT DEFINITION FILE command generates a USAP variant definition file based on a model SAP variant. The generated definition file is written to standard output.

Note: GENERATE VARIANT DEFINITION FILE requires XBP interface 2.0.

(See Section 2.5 Client Fault Tolerance for information on XBP interface 2.0.)

GENERATE VARIANT DEFINITION FILE Command Line Syntax

Figure 4.37, below, illustrates the command line syntax – using the command line, long form of the configuration options – of the GENERATE VARIANT DEFINITION FILE command.

```
-generate vardef -variant variantname -abapname abapname
```

Figure 4.37 GENERATE VARIANT DEFINITION FILE Command Line Syntax

GENERATE VARIANT DEFINITION FILE Command Argument

The GENERATE VARIANT DEFINITION FILE command can be expressed as:

- -generate Long form

The GENERATE VARIANT DEFINITION FILE command argument, **vardef**, generates a usap variant definition file based on the specified SAP variant. The generated variant definition is printed to standard output.

This command option makes it easy to create complex variant definitions based on pre-existing SAP variants. It also eliminates typing errors that can be introduced by manually coding variant definition files.

GENERATE VARIANT DEFINITION FILE Command Options

Option Name	Description
VARIANT	Pre-existing SAP variant name to use as the model variant.
ABAP_NAME	Name of an ABAP program in an SAP system to which the model variant belongs.

Table 4.32 GENERATE VARIANT DEFINITION FILE Command Options

4.4 Configuration Options

This section identifies the Universal Connector for UNIX configuration options that are not associated with one or more Universal Connector commands. Some of these options are required for every execution of Universal Connector; others are optional for any execution.

(For detailed information on configuration options that are associated with one or more specific commands, see Section 4.3 Commands.)

4.4.1 Configuration Option Categories

The configuration options not associated with one or more specific commands are categorized into logical areas of application, as shown in Table 4.33, below.

The name of each category is a link to the following information in this section:

- Description — Description of the options in the category.
- Options syntax — Syntax of the options on the command line.
- Options — Description of the configuration options in the category and a link to detailed information about those options in the Universal Connector 4.1.0 Reference Guide.

Option Categories	Description
Required	
HOST Options	Specifies the SAP host to which a connection should be made.
USER Options	Identifies the SAP user account with which the command executes.
Optional	
CFT Options	Configures client fault tolerant connection.
COMMAND FILE Options	Specifies an additional source of command options.
EVENT Options	Specifies USAP options required for event generation.
Exit Codes	Requests information pertaining to the USAP program.
INSTALLATION Options	Specifies USAP options required for installation.
LOCAL Options	Specifies USAP options required for local broker registration.
MESSAGE Options	Requests information pertaining to the USAP program.
RFC Options	Configures fault tolerant RFC connection.

Table 4.33 Universal Connector for UNIX - Configuration Option Categories

4.4.2 HOST Options

The HOST options are required to establish a connection with an SAP system.

HOST Options Syntax

Figure 4.38, below, illustrates the command line syntax – using the command line, long form of the configuration options – of the HOST options.

```
{-dest destination | -ashost hostname -sysnr number}
-client client
[-max_xbp version]
[-saplang language]
[-xmiaudit {0|1|2|3}]
```

Figure 4.38 HOST Options Syntax

HOST Options

Option Name	Description
AS_HOST	Host name of an SAP application server.
DESTINATION	Name of a destination defined in the **saprfc.ini** file.
CLIENT	SAP client number.
LOGON_LANGUAGE	SAP logon language used for the USAP session.
MAX_XBP	Maximum version of the SAP XBP interface that will be used during USAP execution.
SYSTEM_NUMBER	SAP system number of an SAP application server.
XMI_AUDIT_LEVEL	Sets the XMI audit level to be used for the execution of the command.

Table 4.34 HOST Options

4.4.3 USER Options

The USER options are required to establish an RFC connection to an SAP system. They establish the SAP user identity.

USER Options Syntax

Figure 4.39, below, illustrates the command line syntax – using the command line, long form of the configuration options – of the USER options.

```
-userid userid
-pwd password
```

Figure 4.39 USER Options Syntax

USER Options

Option Name	Description
USER_ID	SAP user ID with which to logon to the SAP system.
PASSWORD	Password for the SAP user ID.

Table 4.35 USER Options

4.4.4 CFT Options

The CFT options are used to configure a client fault tolerant job run. Client fault tolerance is requested for a USAP job run by specifying a COMMAND_ID.

CFT Options Syntax

Figure 4.40, below, illustrates the command line syntax – using the command line, long form of the configuration options – of the CFT options.

```
-cmdid id
-restart {yes|no|auto}
-autorestartok {yes|no}
-cft_secure_cft {yes|no}
-cft_abap abap_program
-cft_target_host host
-cft_cmd_prefix command_prefix
```

Figure 4.40 CFT Options Syntax

CFT Options

Option Name	Description
ALLOW_AUTO_RESTART	Specification for whether or not a **RESTART** value of **AUTO** will be allowed.
CFT_ABAP_PROGRAM	ABAP program to use for the command ID job step.
CFT_COMMAND_PREFIX	In pre-XBP 2.0 CFT mode, the prefix command required for the operating system of the target host.
CFT_TARGET_HOST	In pre-XBP 2.0 CFT mode, the target host to use for the command ID job step when the command ID option is used.
COMMAND_ID	Identifier used to identify the unit of work represented by a USAP command and the associated SAP job.
RESTART	Specification for whether or not this execution of USAP is a restart of a previous client fault tolerant USAP command.
SECURE_CFT	Mode of client fault tolerance to be used for the command invocation.

Table 4.36 CFT Options

4.4.5 COMMAND FILE Options

The COMMAND FILE options are used to specify a file as a source of configuration options used for a command execution. The options read from a command file are processed exactly like options from any other input source.

Encrypted command files are an excellent place to store sensitive data such as user IDs and passwords. Use the Universal Encrypt utility to encrypt a plain text command file. (For information on Universal Encrypt, see the Universal Products Utilities 4.1.0 User Guide.)

Note: All options, including required and command-specific options, can be placed in a command file.

USAP can process both plain text and encrypted command files. Either type of file can be used, but not both. If both are specified, the plain text file will be used.

COMMAND FILE (Plain Text) Options Syntax

Figure 4.41, below, illustrates the command line syntax – using the command line, long form of the configuration options – of the COMMAND FILE (Plain Text) options.

```
-file [filename]
```

Figure 4.41 COMMAND FILE (Plain Text) Options Syntax

COMMAND FILE (Plain Text) Options

Option Name	Description
FILE_NAME	Name of a plain text command file.

Table 4.37 COMMAND FILE (Plain Text) Options

COMMAND FILE (Encrypted) Options Syntax

Figure 4.42, below, illustrates the command line syntax – using the command line, long form of the configuration options – of the COMMAND FILE (Encrypted) options.

```
-encryptedfile [filename [-key key] ]
```

Figure 4.42 COMMAND FILE (Encrypted) Options Syntax

COMMAND FILE (Encrypted) Options

Option Name	Description
ENCRYPT_FILE	Name of an encrypted command file.
ENCRYPTION_KEY	Key used to encrypt the command file.

Table 4.38 COMMAND FILE (Encrypted) Options

4.4.6 EVENT Options

The EVENT options are required for event generation.

Note: EVENT options can be specified only in the configuration file. They have no command line or environment variable parameters.

EVENT Options Syntax

Figure 4.43, below, illustrates the configuration file syntax of the EVENT options.

```
activity_monitoring {yes|no}
event_generation types
```

Figure 4.43 EVENT Options Syntax

EVENT Options

Option Name	Description
ACTIVITY_MONITORING	Specification for whether or not product activity monitoring events are generated.
EVENT_GENERATION	Events to be generated as persistent events.

Table 4.39 EVENT Options

4.4.7 INFORMATIONAL Options

The INFORMATIONAL options request information pertaining to the USAP program.

INFORMATIONAL Options Syntax

Figure 4.44, below, illustrates the command line syntax – using the command line, long form of the configuration options – of the INFORMATIONAL options.

```
-help
-version
```

Figure 4.44 INFORMATIONAL Options Syntax

INFORMATIONAL Options

Option Name	Description
HELP	Writes command line help.
VERSION	Writes USAP version and copyright information.

Table 4.40 INFORMATIONAL Options

4.4.8 INSTALLATION Options

The INSTALLATION options are required for product installation.

Note: INSTALLATION options can be specified only in the configuration file. They have no command line or environment variable parameters.

INSTALLATION Options Syntax

Figure 4.45, below, illustrates the configuration file syntax of the EVENT options.

```
installation_directory directory
```

Figure 4.45 INSTALLATION Options Syntax

INSTALLATION Options

Option Name	Description
INSTALLATION_DIRECTORY	Location in which USAP is installed.

Table 4.41 INSTALLATION Options

4.4.9 LOCAL Options

The LOCAL options are required for local broker registration.

LOCAL Options Syntax

Figure 4.46, below, illustrates the command line syntax – using the command line, long form of the configuration options – of the LOCAL options.

```
-bif_directory directory
-plf_directory directory
```

Figure 4.46 LOCAL Options Syntax

LOCAL Options

Option Name	Description
BIF_DIRECTORY	Broker Interface File (BIF) directory where the Universal Broker interface file is located.
PLF_DIRECTORY	Program Lock File (PLF) directory where the program lock files are located.

Table 4.42 LOCAL Options

4.4.10 MESSAGE Options

The MESSAGE options specify different characteristics of **usap** messages.

MESSAGE Options Syntax

Figure 4.47, below, illustrates the command line syntax – using the command line, long form of the configuration options – of the MESSAGE options.

```
-lang language
-level {trace|audit|info|warn|error}
-trace_file_lines lines
-trace_table size,condition
```

Figure 4.47 MESSAGE Options Syntax

MESSAGE Options

Option Name	Description
MESSAGE_LANGUAGE	Language in which messages are written.
MESSAGE_LEVEL	Level of messages to be written.
TRACE_FILE_LINES	Maximum number of lines to write to the trace file.
TRACE_TABLE	Trace table size and under what conditions it is written to a file.

Table 4.43 MESSAGE Options

4.4.11 RFC Options

The RFC options are always used to configure a fault tolerant RFC connection. All RFC options have default values that are used if additional values are not provided.

RFC Options Syntax

Figure 4.48, below, illustrates the command line syntax – using the command line, long form of the configuration options – of the RFC options.

```
-rfc_logon_retry_interval interval
-rfc_logon_retry_count count
-rfc_listen_interval interval
-rfc_timeout interval
-rfc_retry_interval interval
-rfc_retry_count count
```

Figure 4.48 RFC Options Syntax

RFC Options

Option Name	Description
LOGON_RETRY_INTERVAL	Number of seconds that will elapse between a failed RFC logon attempt and the retry of that logon attempt.
LOGON_RETRY_COUNT	Number of unsuccessful RFC logon retry attempts that can occur before USAP terminates the logon process and ends unsuccessfully.
LISTEN_INTERVAL	Number of seconds that will elapse between RFC listen calls.
TIMEOUT_INTERVAL	Number of seconds that can elapse before USAP considers an RFC call to have timed out.
RETRY_CALL_INTERVAL	Number of seconds that will elapse between a failed RFC call and the retry of that call.
RETRY_CALL_COUNT	Number of unsuccessful RFC call retry attempts that can occur before USAP terminates the RFC call retry process and ends unsuccessfully.

Table 4.44 RFC Options

4.5 Exit Codes

The exit code of **usap** depends on the command being issued.

The following sections identify the exit codes for the various USAP commands.

Note: The default values listed for the exit codes are the installed (configuration file) values. These values may be different than the internal default values (see the Exit Code options in the Universal Connector 4.1.0 Reference Guide).

4.5.1 WAIT for JOB Exit Codes

If the WAIT for JOB command is specified, USAP will map the job's status upon completion to the user definable job exit code parameters.

Table 4.45, below, illustrates this mapping; USAP default values are listed in parentheses.

Job Completion Status in SAP	Exit Code
Terminated	terminated_exit_code (8)
Finished	finished_exit_code (0)
Unknown	22
Error in USAP processing (see Table 4.49).	> 200

Table 4.45 UNIX Wait for Job Exit Codes

4.5.2 WAIT for FS JOB NETWORK Exit Codes

If the WAIT for FS JOB NETWORK command is specified, USAP will map the job network's return code pair to the user definable job network return code parameters. In this case, the exit codes are hard coded and the return code pairs used in the matching process are user definable.

Table 4.46, below, illustrates this mapping; USAP default values are listed in parentheses.

Job network return code pairs used for matching	Exit Code
job_net_rc_00 (02,00;02,02)	0
job_net_rc_04 (02,02)	4
job_net_rc_08 (02,04)	8
job_net_rc_16 (07,00;04,00;02,08)	16
Error in USAP processing (see Table 4.49).	> 200

Table 4.46 UNIX Wait for FS Job Network Exit Codes

4.5.3 DISPLAY STATUS Exit Codes

If the DISPLAY STATUS command is specified, USAP will map the job's current status to the user definable job exit code parameters.

Table 4.47, below, illustrates this mapping; USAP default values are listed in parentheses.

Job Completion Status in SAP	Exit Code
Active	active_exit_code (10)
Ready	ready_exit_code (12)
Scheduled	scheduled_exit_code (14)
Released	released_exit_code (16)
Terminated	terminated_exit_code (8)
Finished	finished_exit_code (0)
Unknown	22
Error in USAP processing (see Table 4.49).	> 200

Table 4.47 UNIX DISPLAY STATUS Exit Codes

4.5.4 DISPLAY QSTATE Exit Codes

If the DISPLAY QSTATE command is specified, USAP will map the queue's current state to the user definable **`qstate`** exit code parameters.

Table 4.48, below, illustrates this mapping; USAP default values are listed in parentheses.

Queue State	Exit Code
'C' to be created	qtobecreated_exit_code (14)
' ' unprocessed	qunprocessed_exit_code (12)
'S' in background	qinbackground_exit_code (10)
'E' error	qerror_exit_code (8)
'F' finished	qfinished_exit_code (0)
Undefined	20
Error in USAP processing (see Table 4.49).	> 200

Table 4.48 UNIX Queue State Exit Codes

4.5.5 All Other Command Exit Codes

If USAP is not performing the WAIT for JOB, WAIT for FS JOB NETWORK, DISPLAY STATUS, or DISPLAY QSTATE command, the exit code indicates the success of the requested actions.

Table 4.49, below, lists the USAP exit codes.

Description	Exit Code
Successfully completed all requested actions.	0
An error occurred processing the requested actions. Messages are printed providing details about the error.	201
Indicates an error with product configuration options or command line options.	210
An error occurred in the initialization phase of message processing. It is possible the error prohibited messages from printing.	211

Table 4.49 USAP for UNIX - Exit Codes

Chapter 5
Job Definition Files

5.1 Overview

This chapter provides information on Universal Connector job definition files, which contain statements that specify the attributes of jobs. These job definitions are used by the SUBMIT, MODIFY, and RUN commands to define or modify jobs in an SAP system.

USAP supports several different job types. The following is a detailed description of the syntax options and requirements for each type of job definition.

5.2 Standard USAP Job Definition File Syntax

The standard USAP job is equivalent to defining a background job SAP via transaction SM36. There are four types of statements used to define a standard USAP job:

1. Job Header statement
2. ABAP Step statement
3. External Step statement
4. External Command Step statement

A job definition requires a Job Header statement followed by one or more ABAP Step statements. Statements are made up of keyword = value assignments and are terminated with a semi-colon (**;**). Each statement type has a specific unique keyword that is required to start the keyword = value assignment list.

Figure 5.1, below., illustrates the syntax of a USAP standard job definition.

```
Job_Header_Statement Step_Statement [Step_Statements]
```

Figure 5.1 USAP Standard Job Definition Syntax

5.2.1 Keywords

The following tables list the keywords available for each statement, the maximum length of the associated values, whether or not they are required, and any restricted value sets.

The first keyword in each table is the keyword required to start the corresponding statement.

5.3 Keywords for Job Header Statement

Table 5.1, below, identifies the keywords for a Job Header statement.

Keyword	Length	Required	Restricted Values / Description
JOBNAME	32	Yes	
TARGET_SERVER	20	No	
JOBCOUNT	8	No	This keyword is useful only for the modify command. In all other cases, it is ignored; it will not cause a syntax error.
JOB_CLASS	1	No	A, B, C This keyword is only valid if at least one of the following requirements is met: 1. Job definition file is being used with an SAP 46C system with support package SAPKB46C44. 2. Job definition file is being used with an SAP 610 system with support package SAPKB61033. 3. Job definition file is being used with an SAP 620 system with support package SAPKB62023. 4. USAP CM interface has been licensed. If the target SAP system meets requirements 1, 2. or 3, USAP will not attempt to use the CM interface. Otherwise, using this keyword with the Auto CM option turned on will cause USAP to use the CM interface for this job.
The following keywords represent SAP job start conditions.			
SDLSTRTDT	8	No	YYYYMMDD
SDLSTRTTM	6	No	HHMMSS
LASTSTRTDT	8	No	YYYYMMDD
LASTSTRTTM	6	No	HHMMSS
PREDJOB	32	No	
PREDJOBCNT	8	No	
EVENTID	32	No	
EVENTPARM	64	No	
CHECKSTAT	1	No	• 'X' = Check job status for subsequent job start. • ' ' = Do not check job status for subsequent job start.
PERIODIC	1	No	• 'X' = Job is periodic. • ' ' = Job is not periodic.
CALENDARID	2	No	
PRDMINS	2	No	00-99
PRDHOURS	2	No	00-99
PRDDAYS	3	No	00-999
PRDWEEKS	2	No	00-99
PRDMONTHS	2	No	00-99
WDAYNO	2	No	00-99

Keyword	Length	Required	Restricted Values / Description
WDAYCDIR	1	No	Work day relative to: • '1' = Beginning of month. • '2' = End of month.
PRDBEHAV	1	No	Start Date Restrictions: • ' ' = Always execute job. • 'D' = Do not execute job on Sundays or holidays. • 'B' = Move job to previous day. • 'A' = Move job to next work day.
NOTBEFORE	8	No	YYYMMDD
The following keywords represent SAP spool list recipient.			
LOGSYS	10	No	
OBJTYPE	10	No	
OBJKEY	70	No	
DESCRIBE	10	No	

Table 5.1 Job Header Keywords

5.4 Keywords for ABAP Step Statement

Table 5.2, below, identifies the keywords for an ABAP Step statement.

Keyword	Length	Required	Restricted Values / Description
ABAP_STEP	40	Yes	
STEP_NUMBER	8	Yes	This keyword is only required and useful for the MODIFY command. It is ignored in all other cases and will not cause a syntax error.
ABAP_PROGRAM_NAME	40	Yes	
SAP_USER_NAME	12	No	
LANGUAGE	1	No	
VARIANT_NAME	14	No	
USERPARAM	250	No	String of variable variant parameters and their associated values; param1=value1;param2=value2;… This keyword is valid only if the USAP CM interface has been licensed. Using this keyword with the Auto CM option turned on will cause USAP to use the CM interface for this job.
The following keywords represent SAP printing parameters.			
OUTPUT_DEVICE	4	No	
PRINT_IMMEDIATELY	1	No	• 'X' = Output. • ' ‘ = Do not output.
RELEASE	1	No	• 'X' = Delete after output.
COPIES	3	No	
ARCHIVING_MODE	1	No	• '1' = only print the document. • '2' = only archive the document. • '3' = both print and archive the document.
SAP_BANNER	1	No	• ' ‘ = no cover sheet. • 'X' = output cover sheet. • 'D' = cover sheet output depends on the setting of the output device (printer) being used.
BANNER_PAGE	1	No	• ‘ ‘ = no cover sheet. • ‘X’ = output cover sheet.
EXPIRATION	1	No	
RECIPIENT	12	No	
NUM_LINES	10	No	
NUM_COLUMNS	10	No	
AUTHORIZATION	12	No	

Keyword	Length	Required	Restricted Values / Description
PLIST	12	No	Spool Request Name This keyword is valid only if the XBP 2.0 interface is installed on the target SAP system or, the USAP CM interface has been licensed. The XBP 2.0 interface will be used if present. Otherwise, if the Auto CM option turned on, USAP will attempt to use the CM interface for this job.
PRTXT	68	No	Spoollist Title This keyword is only valid if the XBP 2.0 interface is installed on the target SAP system or, the USAP CM interface has been licensed. The XBP 2.0 interface will be used if present. Otherwise, if the Auto CM option turned on, USAP will attempt to use the CM interface for this job.
PRNEW	1	No	New Spool Request: • 'X' = Create a new spoollist for each spoollist generated. • ' ' = Append all spoollists. This keyword is valid only if the XBP 2.0 interface is installed on the target SAP system or, the USAP CM interface has been licensed. The XBP 2.0 interface will be used if present. Otherwise, if the Auto CM option turned on, USAP will attempt to use the CM interface for this job.
PRABT	12	No	Department This keyword is valid only if the XBP 2.0 interface is installed on the target SAP system or, the USAP CM interface has been licensed. The XBP 2.0 interface will be used if present. Otherwise, if the Auto CM option turned on, USAP will attempt to use the CM interface for this job.
PAART	16	No	Print format This keyword is only valid if the XBP 2.0 interface is installed on the target SAP system or, the USAP CM interface has been licensed. The XBP 2.0 interface will be used if present. Otherwise, if the Auto CM option turned on, USAP will attempt to use the CM interface for this job.

Keyword	Length	Required	Restricted Values / Description
PRDSN	6	No	Spool Data Set This keyword is only valid if the XBP 2.0 interface is installed on the target SAP system or, the USAP CM interface has been licensed. The XBP 2.0 interface will be used if present. Otherwise, if the Auto CM option turned on, USAP will attempt to use the CM interface for this job.
PTYPE	12	No	Spool Request Type This keyword is only valid if the XBP 2.0 interface is installed on the target SAP system or, the USAP CM interface has been licensed. The XBP 2.0 interface will be used if present. Otherwise, if the Auto CM option turned on, USAP will attempt to use the CM interface for this job.
FOOTL	1	No	Footer: • 'X' = yes. • ' ' = no. This keyword is only valid if the XBP 2.0 interface is installed on the target SAP system or, the USAP CM interface has been licensed. The XBP 2.0 interface will be used if present. Otherwise, if the Auto CM option turned on, USAP will attempt to use the CM interface for this job.
The following keywords represent SAP archiving parameters.			
SAP_OBJECT	10	No	Object Type
AR_OBJECT	10	No	Document Type
INFO	3	No	Info Field
ARCHIV_ID	2	No	Target Storage System This keyword is only valid if the XBP 2.0 interface is installed on the target SAP system or, the USAP CM interface has been licensed. The XBP 2.0 interface will be used if present. Otherwise, if the Auto CM option turned on, USAP will attempt to use the CM interface for this job.
DOC_TYPE	20	No	Document Class This keyword is only valid if the XBP 2.0 interface is installed on the target SAP system or, the USAP CM interface has been licensed. The XBP 2.0 interface will be used if present. Otherwise, if the Auto CM option turned on, USAP will attempt to use the CM interface for this job.

Keyword	Length	Required	Restricted Values / Description
RPC_HOST	32	No	RPC Host This keyword is only valid if the XBP 2.0 interface is installed on the target SAP system or, the USAP CM interface has been licensed. The XBP 2.0 interface will be used if present. Otherwise, if the Auto CM option turned on, USAP will attempt to use the CM interface for this job.
RPC_SERVIC	32	No	RPC Service / RFC Destination This keyword is only valid if the XBP 2.0 interface is installed on the target SAP system or, the USAP CM interface has been licensed. The XBP 2.0 interface will be used if present. Otherwise, if the Auto CM option turned on, USAP will attempt to use the CM interface for this job.
AR_INTERFACE	14	No	Communication Component This keyword is only valid if the XBP 2.0 interface is installed on the target SAP system or, the USAP CM interface has been licensed. The XBP 2.0 interface will be used if present. Otherwise, if the Auto CM option turned on, USAP will attempt to use the CM interface for this job.
MANDANT	3	No	Client This keyword is only valid if the XBP 2.0 interface is installed on the target SAP system or, the USAP CM interface has been licensed. The XBP 2.0 interface will be used if present. Otherwise, if the Auto CM option turned on, USAP will attempt to use the CM interface for this job.
REPORT	40	No	Report Name This keyword is only valid if the XBP 2.0 interface is installed on the target SAP system or, the USAP CM interface has been licensed. The XBP 2.0 interface will be used if present. Otherwise, if the Auto CM option turned on, USAP will attempt to use the CM interface for this job.
ARCTEXT	40	No	Text Information This keyword is only valid if the XBP 2.0 interface is installed on the target SAP system or, the USAP CM interface has been licensed. The XBP 2.0 interface will be used if present. Otherwise, if the Auto CM option turned on, USAP will attempt to use the CM interface for this job.

Keyword	Length	Required	Restricted Values / Description
DATUM	8	No	Archiving Date This keyword is only valid if the XBP 2.0 interface is installed on the target SAP system or, the USAP CM interface has been licensed. The XBP 2.0 interface will be used if present. Otherwise, if the Auto CM option turned on, USAP will attempt to use the CM interface for this job.
ARCUSER	12	No	Data Element for User This keyword is only valid if the XBP 2.0 interface is installed on the target SAP system or, the USAP CM interface has been licensed. The XBP 2.0 interface will be used if present. Otherwise, if the Auto CM option turned on, USAP will attempt to use the CM interface for this job.
PRINTER	4	No	Target Printer This keyword is only valid if the XBP 2.0 interface is installed on the target SAP system or, the USAP CM interface has been licensed. The XBP 2.0 interface will be used if present. Otherwise, if the Auto CM option turned on, USAP will attempt to use the CM interface for this job.
FORMULAR	16	No	Output Format This keyword is only valid if the XBP 2.0 interface is installed on the target SAP system or, the USAP CM interface has been licensed. The XBP 2.0 interface will be used if present. Otherwise, if the Auto CM option turned on, USAP will attempt to use the CM interface for this job.
ARCHIVPATH	70	No	Standard Archive Path This keyword is only valid if the XBP 2.0 interface is installed on the target SAP system or, the USAP CM interface has been licensed. The XBP 2.0 interface will be used if present. Otherwise, if the Auto CM option turned on, USAP will attempt to use the CM interface for this job.
PROTOKOLL	8	No	Storage Connection Protocol This keyword is only valid if the XBP 2.0 interface is installed on the target SAP system or, the USAP CM interface has been licensed. The XBP 2.0 interface will be used if present. Otherwise, if the Auto CM option turned on, USAP will attempt to use the CM interface for this job.

Keyword	Length	Required	Restricted Values / Description
VERSION	4	No	Version Number This keyword is only valid if the XBP 2.0 interface is installed on the target SAP system or, the USAP CM interface has been licensed. The XBP 2.0 interface will be used if present. Otherwise, if the Auto CM option turned on, USAP will attempt to use the CM interface for this job.

Table 5.2 ABAP Step Keywords

5.5 Keywords for External Step Statement

Table 5.3, below, identifies the keywords for an External Step statement.

Keyword	Length	Required	Restricted Values / Description
EXTERNAL_STEP	40	Yes	
STEP_NUMBER	8	Yes	This keyword is required (and useful) only for the Modify command. In all other cases, it is ignored and will not cause a syntax error.
PROGRAM_NAME	128	Yes	
PROGRAM_PARAMETERS	255	No	
SAP_USER_NAME	12	No	
TARGET_HOST	32	Yes	
WAIT_FOR_TERMINATION	1	No	• ' ‘ or ‘W’ = Don't wait. • 'X' or ‘C’ = Wait. • ‘E’ = The external program signals its time limitation over a Event to the SAP system.
CONNCNTL	1	No	• ‘R’ = Communication way is held after starting the external program. • ‘H’ = Communication way is diminished after starting the external program.
STDINCNTL	1	No	• ‘N’ = No change. • ‘C’ = Standard input closes. • ‘R’ = Return standard input.
STDOUTCNTL	1	No	• ‘N’ = No change. • ‘C’ = Standard output expenditure closes. • ‘R’ = Return standard output expenditure. • ‘T’ = Return standard output into the trace file. • ‘M’ = Write standard output expenditure into main storage.
STDERRCNTL	1	No	• ‘N’ = No change. • ‘C’ = Standard error expenditure closes. • ‘R’ = Return standard error expenditure. • ‘M’ = Write standard error expenditure into main storage.
TRACECNTL	1	No	• ‘0’ = Level 0, no trace. • ‘1’ = Level 1, function call trace. • ‘2’ = Level 2, minutes trace. • ‘3’ = Level 3, expression of all messages.

Table 5.3 External Step Keywords

5.6 Keywords for External Command Step Statement

Table 5.4, below, identifies the keywords for an External Command Step statement.

Keyword	Length	Required	Restricted Values / Description
COMMAND_STEP	40	Yes	
STEP_NUMBER	8	Yes	This keyword is required (and useful) only for the Modify command. In all other cases, it is ignored and will not cause a syntax error.
COMMAND_NAME	128	Yes	
COMMAND_PARAMETERS	255	No	
SAP_USER_NAME	12	No	
TARGET_HOST	32	Yes	
OPSYSTEM	10	Yes	
WAIT_FOR_TERMINATION	1	No	• ' ‘ or ‘W’ = Don't wait. • 'X' or ‘C’ = Wait. • ‘E’ = The external program signals its time limitation over a Event to the SAP system.
CONNCNTL	1	No	• ‘R’ = Communication way is held after starting the external program. • ‘H’ = Communication way is diminished after starting the external program.
STDINCNTL	1	No	• ‘N’ = No change. • ‘C’ = Standard input closes. • ‘R’ = Return standard input.
STDOUTCNTL	1	No	• ‘N’ = No change. • ‘C’ = Standard output expenditure closes. • ‘R’ = Return standard output expenditure. • ‘T’ = Return standard output into the trace file. • ‘M’ = Write standard output expenditure into main storage.
STDERRCNTL	1	No	• ‘N’ = No change. • ‘C’ = Standard error expenditure closes. • ‘R’ = Return standard error expenditure. • ‘M’ = Write standard error expenditure into main storage.
TRACECNTL	1	No	• ‘0’ = Level 0, no trace. • ‘1’ = Level 1, function call trace. • ‘2’ = Level 2, minutes trace. • ‘3’ = Level 3, expression of all messages.

Table 5.4 External Command Step Keywords

5.7 Sample USAP Job Definition File

Figure 5.2, below, illustrates a sample job definition file that defines a job with 1 ABAP step running ABAP report BTCSPOOL.

```
/* Job Header statement */
JOBNAME                  = "SAMPLE_1"
    JOB_CLASS            = "B"
;

/* ABAP_STEP Step statement */
ABAP_STEP                = "1"
   ABAP_PROGRAM_NAME     = "BTCSPOOL"
   PRTXT                 = "Sample 1"
   PRNEW                 = "X"
;
```

Figure 5.2 USAP Job Definition File

5.8 CM Report Job Definition File Syntax

CM Report jobs are SAP jobs that run a single ABAP report. USAPX submits and starts CM report jobs on the SAP system using CM function modules within SAP. CM report jobs offer several advantages over standard SAP jobs defined using XBP (standard USAP jobs). CM report jobs allow control of job attributes that are not available through XBP. These attributes include job class and certain print and archiving parameters.

A CM report job is defined in a single statement. The statement is made up of keyword = value assignments and is terminated with a semi-colon (;).

Table 5.5, below, lists the keywords available for a CM report job statement, the maximum length of the associated values, whether or not they are required, and any restricted value sets. The first keyword in the table is the keyword required to start the statement.

Keyword	Length	Required	Restricted Values / Description
CM_REPORT	8	Yes	Name of the SAP ABAP report to be started.
CM_OPC_TOKEN	15	Yes	Unique character string that is used to prefix the job created for the processing of the CM report job.
CM_CLASS	1	No	A, B, C
CM_USER	12	No	User that will be used for the authorization check.
CM_VARIANT	14	No	Name of the variant that will be used for processing the ABAP report.
CM_USERPARAM	250	No	String of variable variant parameters and their associated values; param1=value1;param2=value2;param3=value3; param4=value4...
The following keywords represent SAP printing parameters.			
PDEST	4	No	Output Device
PRCOP	3	No	Number of Copies
PLIST	12	No	Spool Request Name
PRTXT	68	No	Spoollist Title
PRIMM	1	No	Print Immediately: • 'X' = Output. • ' ‘ = Do not output.
PRREL	1	No	Delete After Print: • 'X' = Delete after output. • ‘ ‘ = Do not delete after output.
PRNEW	1	No	New Spool Request: • ‘X’ = Create a new spoollist for each spoollist generated as a result of processing the CM communication. • ‘ ‘ = Append all spoollists generated as a result of processing the CM communication.
Keyword	Length	Required	Restricted Values
PEXPI	1	No	Retention Period
LINCT	10	No	Lines per page

Keyword	Length	Required	Restricted Values / Description
LINSZ	10	No	Columns per page
PAART	16	No	Print format
PRBIG	1	No	Selection Cover Sheet
PRSAP	1	No	SAP Cover Sheet
PRREC	12	No	Recipient
PRABT	12	No	Department
PRBER	12	No	Authorization
PRDSN	6	No	Spool Data Set
PTYPE	12	No	Spool Request Type
ARMOD	1	No	Archiving Mode: • '1' = only print the document. • '2' = only archive the document. • '3' = both print and archive the document.
FOOTL	1	No	Footer
The following keywords represent SAP archiving parameters.			
SAP_OBJECT	10	No	Object Type
AR_OBJECT	10	No	Document Type
INFO	3	No	Info Field
ARCHIV_ID	2	No	Target Storage System
DOC_TYPE	20	No	Document Class
RPC_HOST	32	No	RPC Host
RPC_SERVIC	32	No	RPC Service / RFC Destination
AR_INTERFACE	14	No	Communication Component
MANDANT	3	No	Client
REPORT	40	No	Report Name
ARCTEXT	40	No	Text Information
DATUM	8	No	Archiving Date
ARCUSER	12	No	Data Element for User
PRINTER	4	No	Target Printer
FORMULAR	16	No	Output Format
ARCHIVPATH	70	No	Standard Archive Path
PROTOKOLL	8	No	Storage Connection Protocol
VERSION	4	No	Version Number

Table 5.5 CM Report Job Keywords

5.9 CM Communication Job Definition File Syntax

A CM communication is defined in a single statement. The statement is made up of keyword = value assignments and is terminated with a semi-colon (;).

The following tables list the keywords available for a CM Communication statement, the maximum length of the associated values, whether or not they are required, and any restricted value sets. The first keyword in the table is the keyword required to start the statement.

Table 5.6, below, identifies the keywords for a CM Communication statement.

Keyword	Length	Required	Restricted Values / Description
CM_COMMUNICATION_ID	12	Yes	Name of CM communication to be started.
CM_OPC_TOKEN	15	Yes	Unique character string that is used to prefix jobs created during the processing of a CM communication.
CM_MODE	1	No	Starting mode for CM communication: • -N = Normal (Start with the 1. Component). • -R = Restart (with the last incorrect component). • -S = Skip (Start after the last incorrect component). • -T = All run time information delete. Default = -N.
CM_SPEC	1	No	A value of 'V' will prevent the CM communication processing jobs from being prefixed with the CM_OPC_TOKEN value. Therefore, the CM communication processing jobs will be created using only the CM_COMMUNICATION_ID value.
CM_USER	12	No	User that will be used for the authorization check.
CM_JOBLOG	1	No	Controls the return of the SAP joblog for processed communication components: • 0 = no information. • 1 = only in the event of an error. • 2 = always. Default = 1.
CM_SPOOL	1	No	Controls the return of the SAP spoollist for processed communication components: • 0 = no information. • 1 = only in the event of an error. • 2 = always. Default = 1.
The following keywords represent SAP printing parameters.			
PDEST		No	Output Device
PRCOP		No	Number of Copies
PLIST		No	Spool Request Name
PRTXT		No	Spoollist Title
PRIMM		No	Print Immediately: • 'X' = Output. • ' ' = Do not output.

Keyword	Length	Required	Restricted Values / Description
PRREL		No	Delete After Print: • 'X' = Delete after output. • ' ' = Do not delete after output.
PRNEW		No	New Spool Request: • 'X' = Create a new spoollist for each spoollist generated as a result of processing the CM communication. • ' ' = Append all spoollists generated as a result of processing the CM communication.
PEXPI		No	Retention Period
LINCT		No	Lines per page
LINSZ		No	Columns per page
PAART		No	Print format
PRBIG		No	Selection Cover Sheet
PRSAP		No	SAP Cover Sheet
PRREC		No	Recipient
PRABT		No	Department
PRBER		No	Authorization
PRDSN		No	Spool Data Set
PTYPE		No	Spool Request Type
ARMOD		No	Archiving Mode: • '1' = only print the document. • '2' = only archive the document. • '3' = both print and archive the document.
FOOTL		No	Footer
The following keywords represent SAP archiving parameters.			
SAP_OBJECT		No	Object Type
AR_OBJECT		No	Document Type
INFO		No	Info Field
ARCHIV_ID		No	Target Storage System
Keyword	Length	Required	Description / Restricted Values
DOC_TYPE		No	Document Class
RPC_HOST		No	RPC Host
RPC_SERVIC		No	RPC Service / RFC Destination
AR_INTERFACE		No	Communication Component
MANDANT		No	Client
REPORT		No	Report Name
ARCTEXT		No	Text Information
DATUM		No	Archiving Date
ARCUSER		No	Data Element for User
PRINTER		No	Target Printer
FORMULAR		No	Output Format

Keyword	Length	Required	Restricted Values / Description
ARCHIVPATH		No	Standard Archive Path
PROTOKOLL		No	Storage Connection Protocol
VERSION		No	Version Number

Table 5.6 CM Communication Statement Keywords

5.10 Variant Definition File

USAP variant definition files contains statements that specify the attributes of variants. These variant definitions are used by the SUBMIT and MODIFY commands to define or modify variants in an SAP system. The following is a detailed description of the syntax options and requirements for variant definition files.

5.10.1 Variant Definition File Syntax

The USAP variant definition file is used to create or modify a variant in an SAP system. There are three types of statements used to define a variant:

1. Variant Header Statement
2. Variant Text Statement
3. Variant Content Statement

A variant definition requires a Variant Header statement followed by Text and Content statements. Statements are made up of keyword = value assignments and are terminated with a semi-colon (;). Each statement type has a specific unique keyword that is required to start the keyword = value assignment list.

Figure 5.3, below, illustrates the syntax of a USAP variant definition.

```
Variant_Header_Statement Variant_Text_Statement Variant_Content_Statement
[Variant_Content_Statements]
```

Figure 5.3 USAP Variant Definition Syntax

5.10.2 Keywords

The following tables list the keywords available for each statement, the maximum length of the associated values, whether or not they are required, and any restricted value sets. The first keyword in each table is the keyword required to start the corresponding statement.

Keywords for Variant Header Statement

Table 5.7, below, identifies the keywords for a Variant Header statement.

Keyword	Length	Required	Restricted Values
VARIANT_NAME	14	Yes	Variant name.
REPORT	40	Yes	ABAP report for which variant is defined.

Table 5.7 Variant Header Keywords

Keywords for Variant Text Statement

Table 5.8, below, identifies the keywords for a Variant Text statement.

Keyword	Length	Required	Restricted Values
VARIANT_TEXT	30	Yes	
LANGUAGE	2	Yes	

Table 5.8 Variant Text Step Keywords

Keywords for Variant Content Statement

Table 5.9, below, identifies the keywords for a Variant Content statement.

Keyword	Length	Required	Restricted values
SELNAME	8	Yes	
KIND	1	Yes	Field type: • 'P' = Field type is a parameter. • 'S' = Field type is a selection option.
SIGN	1	Yes	Selection sign: • 'I' = Include values based on field selection criteria. • 'E' = Exclude values based on field selection criteria.
OPTION	2	Yes	Selection option: • 'CP' = Pattern. • 'EQ' = Single value. • 'GE' = Greater than or equal to. • 'LE' = Less than or equal to. • 'GT' = Greater than. • 'LT' = Less than. • 'NE' = Not equal to.
LOW	45	No	Selection value.
HIGH	45	No	Selection value.
PROTECTED	1	No	• 'X' = Field is protected. • ' ' = Field is not protected.
APPENDAGE	1	No	• 'X' = Appendage. • ' ' = Not appendage.
VNAME	30	No	Name of variant variable.
VTYPE	1	No	Variant variable type: • 'T' = Table variable from TVARV. • 'D' = Dynamic date calculation. • 'B' = User defined variables.
INVISIBLE	1	No	Hide field: • 'X' = Invisible. • ' ' = Not invisible.
NOINT	1	No	Hide field 'BIS': • 'X' = Invisible. • ' ' = Not invisible.
SCREENNR	4	No	Screen number.
NO_IMPORT	1	No	Save field without values: • 'X' = Yes • ' ' = No
OBLI	1	No	Required field: • 'X' = Yes • ' ' = No

Table 5.9 Variant Content Step Keywords

5.10.3 Sample Variant Definition File

Figure 5.4, below, illustrates a sample variant definition file.

This file will define variant **SAMPLE_1** for ABAP report **RSUSR002**. The **USER** field will contain value **S***.

```
/* Variant Header statement. */
VARIANT_NAME   = "SAMPLE_1"
   REPORT      = "RSUSR002"
;

/* Variant text statement. */
VARIANT_TEXT   = "SAMPLE_1"
   LANGUAGE    = "EN"
;

/* User */
SELNAME        = "USER"
   KIND        = "S"
   SIGN        = "I"
   OPTION      = "CP"
   LOW         = "S*"
   HIGH        = ""
   PROTECTED   = ""
   APPENDAGE   = ""
   VNAME       = ""
   VTYPE       = ""
   INVISIBLE   = ""
   SCREENNR    = ""
   NO_IMPORT   = ""
   SPAGPA      = ""
   OBLI        = ""
   NOINT       = ""
;
```

Figure 5.4 Variant Definition File (Sample)

5.11 Job Intercept Table Definition File

USAP job intercept table definition files contains statements that specify criteria rows. These definitions are used by the SUBMIT command to replace or append the job intercept table in an SAP system.

The following is a detailed description of the syntax options and requirements for job intercept table definition files.

5.11.1 Job Intercept Table Definition File Syntax

The USAP job intercept table definition file is used to replace or append the job intercept in an SAP system.

There are two types of statements used to define a job intercept table:

1. Job Intercept Table Header Statement
2. Job Intercept Table Row Statement

A job intercept table definition requires a Header statement followed by row statements. Statements are made up of keyword = value assignments and are terminated with a semi-colon (;). Each statement type has a specific unique keyword that is required to start the keyword = value assignment list.

Figure 5.5, below, illustrates the syntax of a USAP job intercept table definition.

```
Job_Intercept_Table_Header_Statement  [Job_Intercept_Table_Row_Statements]
```

Figure 5.5 USAP Job Intercept Table Definition Syntax

5.11.2 Keywords

The following tables list the keywords available for each statement, the maximum length of the associated values, whether or not they are required, and any restricted value sets. The first keyword in each table is the keyword required to start the corresponding statement.

Keywords for Job Intercept Table Header Statement

Table 5.10, below, identifies the keywords for a Job Intercept Table Header statement.

Keyword	Length	Required	Restricted values
INTERCEPT_TABLE	1024	Yes	Table name **This value is only used internally by USAP. It does not effect the SAP table definition.
APPEND	1	Yes	• 'X' = Append. • ' ' = Replace.

Table 5.10 Job Intercept Table Header Keywords

Keywords for Job Intercept Table Row Statement

Table 5.11, below, identifies the keywords for a Job Intercept Table Row statement.

Keyword	Length	Required	Restricted Values
INTERCEPT_ROW	1024	Yes	Row name **This value is only used internally by USAP. It does not effect the SAP table definition.
CLIENT	3	No	
JOB_NAME	32	No	
JOB_CREATOR	12	No	

Table 5.11 Job Intercept Row Step Keywords

5.11.3 Sample Job Intercept Table Definition File

Figure 5.6, below, illustrates a sample job intercept table definition file.

The file will append 4 rows to the SAP job intercept criteria table.

```
/* Job Intercept Table Header statement */
INTERCEPT_TABLE          = "TABLE_1"
    APPEND               = "X"
;

/* Job Intercept Row statement */
INTERCEPT_ROW            = "1"
    CLIENT            = "850"
    JOB_NAME             = "TEST*"
    JOB_CREATOR          = "stonebranch"
;

/* Job Intercept Row statement */
INTERCEPT_ROW            = "2"
    CLIENT            = "850"
    JOB_NAME             = "TST*"
    JOB_CREATOR          = "stonebranch"
;

/* Job Intercept Row statement */
INTERCEPT_ROW            = "3"
    CLIENT            = "850"
    JOB_NAME             = "DEV*"
    JOB_CREATOR          = "stonebranch"
;

/* Job Intercept Row statement */
INTERCEPT_ROW            = "4"
    CLIENT            = "*"
    JOB_NAME             = "*"
    JOB_CREATOR          = "BOB"
;
```

Figure 5.6 Job Intercept Table Definition File (Sample)

5.12 FS Job Network Definition File

USAP FS job network definition files contains statements that specify the attributes of FS job networks. These variant definitions are used by the SUBMIT, START, and RUN commands to define and start FS job networks in an SAP system.

The following is a detailed description of the syntax options and requirements for FS job network definition files.

5.12.1 FS Job Network Definition File Syntax

The USAP FS job network definition file is used to create an FS job network in an SAP system.

There are three types of statements used to define an FS job network:

1. FS Jobnet Header statement.
2. FS Jobnet Process statement.
3. FS Jobnet Process Relation statement.

An FS jobnet definition requires an FS Jobnet Header statement followed by FS Jobnet Process Statements, then FS Jobnet Process Relation statements. Statements are made up of keyword = value assignments and are terminated with a semi-colon (**;**). Each statement type has a specific unique keyword that is required to start the keyword = value assignment list.

Figure 5.7, below, illustrates the syntax of a USAP FS Job Network definition.

```
FS_Jobnet_Header FS_Jobnet_Process [FS_Jobnet_Process]
FS_Jobnet_Process_Relation [FS_Jobnet_Process_Relation]
```

Figure 5.7 USAP FS Jobnet Definition Syntax

5.12.2 Keywords

The following tables list the keywords available for each statement, the maximum length of the associated values, whether or not they are required, and any restricted value sets.

The first keyword in each table is the keyword required to start the corresponding statement.

Keywords for FS Jobnet Header Statement

Table 5.12, below, identifies the keywords for an FS Jobnet Header statement.

Keyword	Length	Required	Restricted Values
NETWORKIDENTIFIER	50	Yes	Network identifier.

Table 5.12 FS Jobnet Header Keywords

Keywords for FS Jobnet Process Statement

Table 5.13, below, identifies the keywords for an FS Jobnet Process statement.

Keyword	Length	Required	Restricted Values
PROCESS_IDENTIFIER	50	Yes	Process identifier.
REPORT_NUMBER	3	Yes	Report number: 001-999
REPORT_NAME	40	Yes	Report name.
REPORT_VARIANT	14	No	Report variant.
JOBNAME	32	Yes	Job name.

Table 5.13 FS Jobnet Process Step Keywords

Keywords for FS Jobnet Process Relation Statement

Table 5.14, below, identifies the keywords for an FS Jobnet Process Relation statement.

Keyword	Length	Required	Restricted Values
PROCESS_RELATION	3	Yes	Process relation number: 001-999.
REPORT_NUMBER_PREDECESSOR	3	Yes	Report number predecessor: 001-999.
REPORT_NUMBER_SUCCESSOR	3	Yes	Report number successor: 001-999.

Table 5.14 FS Jobnet Process Relation Step Keywords

Sample FS Job Network Definition File

Figure 5.8, below, illustrates a sample FS job network definition file.

```
/*********************************************************************
**
** Sample FS Job Network definition file for USAP for
** SAP
**
** Demonstrates creation of a multi-process jobnet.
**
*********************************************************************/

/* Jobnet Header statement */
NETWORKIDENTIFIER               = "SB-NETID_01";

/* Add Jobnet Process statements */
PROCESS_IDENTIFIER              = "SB-PRC_01"
    REPORT_NUMBER               = "001"
    REPORT_NAME                 = "Z_TEST_NETWORK"
    REPORT_VARIANT              = "RC_00"
    JOBNAME                     = "SB-Z_TEST_NETWORK"
   ;

PROCESS_IDENTIFIER              = "SB-PRC_02"
    REPORT_NUMBER               = "002"
    REPORT_NAME                 = "RSUSR000"
    JOBNAME                     = "SB-RSUSR000"
 ;
```

```
PROCESS_IDENTIFIER              = "SB-PRC_03"
    REPORT_NUMBER               = "003"
    REPORT_NAME                 = "BTCSPOOL"
    JOBNAME                     = "SB-BTCSPOOL"
;

/* Add Jobnet Process Relations statements. */
PROCESS_RELATION                = "1"
    REPORT_NUMBER_PREDECESSOR   = "001"
    REPORT_NUMBER_SUCCESSOR     = "002"
;

PROCESS_RELATION                = "2"
    REPORT_NUMBER_PREDECESSOR   = "002"
    REPORT_NUMBER_SUCCESSOR     = "003"
;
```

Figure 5.8 FS Job Network Definition File (Sample)

5.13 Spoollist Translation Tables

USAP returns spoollists in a raw (SAP internal) format. This raw format contains all of the formatting control codes that the SAP system needs to display or print the spoollist. In most cases, this raw format will not be desirable. Therefore, USAP provides the ability to translate the raw spoollist into a desirable format. The translation is performed using a user definable translation table. Multiple translation tables can be defined to achieve different formatting results. The required translation table can be specified at run time.

On UNIX systems, the Spoollist Translation (STT) files are located in the NLS subdirectory of the installation directory.

z/OS

The STT files are located in the library allocated to the UNVNLS DD statement.

USAP ships with two Spoollist Translation files: **`default.stt`** and **`raw.stt`**. The default Spoollist Translation Table file is **`default`**. This translation table contains translations for the standard SAP formatting codes to appropriate character representations. The **`raw`** translation table defines no translations and allows USAP to return the spoollist in its SAP internal format.

5.13.1 Spoollist Translation Table File Format

The Spoollist Translation Table files consist of three white space-separated columns.

Column 1

This is a compare string to look for in the raw unformatted spoollist. This compare string is built by combining comma delimited values. The values are combined to make up the actual compare string. The values can be quoted strings, hexadecimal values representing characters, or decimal values representing characters.

Note: Spaces cannot be used to separate values in the comma delimited list.

Column 2

This is a replace string that will be used to replace the compare string in the raw unformatted spoollist. This replace string is built from comma delimited values. The values are combined to make up the actual replace string. The values can be quoted strings, hexadecimal values representing characters, or decimal values representing characters.

Note: Spaces cannot be used to separate values in the comma delimited list.

Column 3

This is a single decimal value used to restrict the comparison to a specific starting column. A value is not required in this column. If no value is specified in this column, the compare string will be replaced in every location that it is found.

5.14 Examples

This section contains examples demonstrating the use of USAP.

The following list provides a link to each example.

- Define a Job Using a Job Definition File
- Define a Job Using an Existing SAP Job
- Define Job, Run Job, Get Output, and Purge Job

5.14.1 Define a Job Using a Job Definition File

This example uses a USAP job definition file to define a job to an SAP system. The job, upon USAP completion, exists in the SAP system in a scheduled state.

```
usap -sub sample1.usp -userid sapuser -pwd sappwd
     -dest BIN_HS0092 -client 800
```

Command Line Options

The command line options used are:

Command Options	Description
`-sub`	Specifies the submit command which defines a job to an SAP system. The job definition file **`sample1.usp`** is used as the source for the job definition.
`-userid`	Specifies the external SAP user ID with which the command is executed.
`-pwd`	Specifies the password for the user ID.
`-dest`	Destination name in the **`saprfc.ini`** file.
`-client`	SAP client number.

5.14.2 Define a Job Using an Existing SAP Job

This example uses an existing job in an SAP system as a model and creates a copy. The job name remains the same and the jobid is generated by the SAP system.

```
usap –sub –j SAMPLE1 –b 10080901 –userid sapuser –pwd sappwd
     –dest BIN_HS0092 –client 800
```

Command Line Options

The command line options used are:

Command Options	Description
`-sub`	Specifies the submit command which defines a job to an SAP system. The lack of a job definition file indicates that the definition will use an existing job as a model. That job will be identified by **`–jobname`** and **`–jobid`**.
`-userid`	Specifies the external SAP user ID with which the command is executed.
`-pwd`	Specifies the password for the user ID.
`-dest`	Destination name in the **`saprfc.ini`** file.
`-client`	SAP client number.

5.14.3 Define Job, Run Job, Get Output, and Purge Job

This example uses an existing job in an SAP system as a model and creates a copy. The newly created job then is started. USAP waits for the job to finish, and then writes the joblog to standard error and the spoollists to standard out. Finally, the job and its output are purged from the SAP system.

```
usap –sub –j SAMPLE1 –b 10080901 –start –wait –purge
–userid sapuser –pwd sappwd –dest BIN_HS0092 –client 800
```

Command Options

The command options used are:

Command Options	Description
`-sub`	Specifies the submit command which defines a job to an SAP system. The lack of a job definition file indicates that the definition will use an existing job as a model. That job will be identified by **`–jobname`** and **`–jobid`**.
`-start`	Specifies that the job should be started.
`-wait`	Causes USAP to wait for the job to complete.
`-purge`	Specifies that the job and its associated output are to be purged from the SAP system.
`-userid`	Specifies the external SAP user ID with which the command is executed.
`-pwd`	Specifies the password for the user ID.
`-dest`	Destination name in the **`saprfc.ini`** file.
`-client`	SAP client number.

Appendix A
Customer Support

Stonebranch, Inc. provides customer support, via telephone and e-mail, for Universal Connector and all Indesca / Infitran components.

E-MAIL

All Locations

support@stonebranch.com

Customer support contact via e-mail also can be made via the Stonebranch website:

www.stonebranch.com

TELEPHONE

Customer support via telephone is available 24 hours per day, 7 days per week.

North America

(+1) 678 366-7887, extension 6

(+1) 877 366-7887, extension 6 [toll-free]

Europe

+49 (0) 700 5566 7887

950 North Point Parkway, Suite 200
Alpharetta, Georgia 30005
U.S.A.