APTRA™Advance NDC

Developer's Guide

B006-6046-J000
Issue 1
July 2007

The product described in this book is a licensed product of NCR Corporation.

APTRA is a trademark of NCR Corporation.
Microsoft, Windows, Windows NT, Windows XP, Visual Studio, ActiveX, and Visual Basic Script are registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.
IBM and OS/2 are trademarks of International Business Machines Corporation.
Diebold is a trademark of Diebold, Incorporated.
Sound Blaster is a trademark of Creative Technology Ltd.
Adobe, Acrobat and Reader are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States and/or other countries.
Autodesk and FLIC are a trademarks of Autodesk, Inc.

Disclaimer:

It is the policy of NCR Corporation to improve products as technology, components, software and firmware become available. NCR therefore reserves the right to change specifications without prior notice.

All features, functions and operations described herein may not be marketed by NCR in all parts of the world. In some instances, photographs are of equipment prototypes. Therefore, before using this document, consult with your NCR representative or NCR office for information that is applicable and current.

To maintain the quality of our publications, we need your comments on the accuracy, clarity, organisation and value of this book.

Address correspondence to:

NCR Financial Solutions Group Ltd
Information Solutions Feedback
Kingsway West
Dundee
Scotland
DD2 3XX

© 2007
By NCR Corporation
Dayton, Ohio, USA
All Rights Reserved

Federal Communications Commission (FCC) Radio Frequency Interference Statement

Note: This equipment has been tested and found to comply with the limits for a Class A digital device, pursuant to Part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference in which case the user will be required to correct the interference at his own expense.

Canadian Class A Device Declaration

This digital apparatus does not exceed the Class A limits for radio noise emissions from digital apparatus set out in the Radio Interference Regulations of the Canadian Department of Communications.

Le présent appareil numérique n'émet pas de bruits radioélectriques dépassant les limites applicables aux appareils numériques de la classe A prescrites dans le Réglement sur le brouillage radioélectrique édicté par le ministère des Communications du Canada.

Information to User

This equipment must be installed and used in strict accordance with the manufacturer's instructions. However, there is no guarantee that interference to radio communications will not occur in a particular commercial installation. If this equipment does cause interference, which can be determined by turning the equipment off and on, the user is encouraged to consult an NCR service representative immediately.

Caution NCR Corporation is not responsible for any radio or television interference caused by unauthorised modifications of this equipment or the substitution or attachment of connecting cables and equipment other than those specified by NCR. Such unauthorised modifications, substitutions, or attachments may void the user's authority to operate the equipment. The correction of interference caused by such unauthorised modifications, substitutions, or attachments will be the responsibility of the user.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Revision Record

Date	Page	Description of Change
July 2007		New revision for Advance NDC 3.02
	All chapters	Restructured to include more configuration, troubleshooting, and exits information and incorporate white papers; Replaced the *APTRA Advance NDC, General Description* with the *APTRA Advance NDC, Overview*; Replaced the *APTRA Advance ADE, User's Guide* with the *APTRA Author User's Guide;* Updated Advance ADE to APTRA Author
	xlvi	Explained that hyperlinks in the revision record are only active for the current release.
	xlvii	Explained that the APTRA Simulator product includes the XFS Simulator
	1-1	Updated reference to the APTRA Advance Application Environment to the APTRA Author environment; Added reference to the *APTRA Advance NDC, Overview* for information on new features.
	1-1 1-16	Removed coin dispense, which is now supported, from DPM information describing differences between NDC+ and Advance NDC
	1-8	Removed reference to CPM and ActiveX, as CPM is now accessed through CEN-XFS
	1-10	Added information about using the general purpose buffers
	1-15	Added new cash dispenser cassette types section
	1-19	Added new coin dispense section describing differences between NDC+ and Advance NDC
	1-23	Added statement that Diebold is unsupported in Advance NDC
	1-25	Updated information about the EJ log file checksum
	1-26	Updated information about the EJ file format

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Date	Page	Description of Change
	1-27	Added section on the use of K screens; Removed coin dispenser from list, as now supported
	1-29	Removed DISP COINS, CLR COINS, ADD COINS, STD COINS, CHECK CDM, SET COINS, TRACE ON, and TRACE OFF from the table of unsupported Supervisor functions
	1-30	Added section on leaving Supervisor and Diagnostics simultaneously
	1-33	Updated reference to font information
	1-34	Added alphanumeric state number information, which allows up to 46656 state numbers to be supported
	1-35	Updated dual cash handler support; Added differences in screen support; Added difference in support for the set display mode control sequence
	1-39	Updated and added enhanced EJ features
	1-40	Added descriptions for Silent Debug and DebugLog.
	1-41	Added description for message reflection; Updated Device Access topic to state that in Advance NDC 3.02 all devices are accessed through the CEN-XFS interface and removed reference to CPM.
	2-1	Added reference to Chapter 10, “Delivering an Advance NDC Application to the SST” and removed old reference to the HTML Package IP.
	2-2 to 2-4	Added references to the *APTRA Advance NDC Overview* document.
	2-4	Reference to Chapter 5 added for Silent Debug and DebugLog.
	2-4, 2-5	Updated de-installation instructions
	3-3	Added section on cardholder graphics
	3-8	Updated the method for changing the *resrvd.def* file
	3-10	Updated to state that all devices are accessed through the CEN-XFS interface and removed reference to CPM.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Date	Page	Description of Change
	3-11	Removed explanation of restriction on outgoing interceptor to reflect messages as this is now supported
	3-13	Updated STCONT format for alphanumeric state numbers
	3-15	Updated the primary and extension state table information for nested states
	3-17, 3-21, and 3-22	Updated reference to Visual C++ with reference to Microsoft Visual Studio 2005
	3-18, 3-21	Added FreeStringVal to the allocate and free memory function signatures table and callback link libraries and headers table
	3-21	Updated header file information
	4-1	Added reference to see Chapter 5 for configuration information.
	4-2	Updated to state that CEN-XFS interface is now used with CPM, not ActiveX; Added statement that enhanced Night Safe is not supported
	4-3	Added note regarding card security settings and Option 46
	4-6	Updated information about the APTRA Simulator
	4-8	Added comment that only basic Night Safe is supported
	4-9	Changed device access entry for CPM in Table 4-2 to PTR SP
	4-9 to 4-11	Added Coin Dispenser to the service provider access details, supplies data source information, and fitness data source information
	4-11	Changed data source entry for CPM in Table 4-4 to XFS PTR
	4-12	Updated the message handling changes summary for the tamper and sensors data
	4-15 to 4-40	Removed configuration information as this is now in Chapter 5.
	4-15	Added new section for STCONT file updates

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Date	Page	Description of Change
	Chapter 5 to 9	Added a new Chapter 5. Previous Chapters 5 to 9 are now Chapters 6 to 10. The new Chapter 5 title is “Configuring Advance NDC and Support Applications”.
	5-1	Added overview information.
	5-2 to 5-62	Added configuration information previously in Chapter 4.
	5-2, 5-6 to 5-8	Updated cash handler and dual cash handler information for increased capacity, option 76, and maximum notes to dispense
	5-4	Added section on setting up seven cassette types
	5-6	Added information on supplies and severity reporting
	5-8	Updated information on the cash handler status message
	5-9	Added interlock handling with dual dispensers Added coin dispenser configuration
	5-12	Added information on clearing the communications buffer; Updated off-line timer
	5-14	Added information on suppressing cash low messages in a dialup environment
	5-17	Amended default statement length setting; Added information on receipt retract; Added registry configuration for open statement form-based printing
	5-20	Added USB receipt and journal printer configuration information
	5-21	Added information on producing *.prn* files for use with USB printers
	5-22	Added Promote coupon printing configuration information Updated Print Track 2 information and added journalling mask configuration
	5-24	Added information on configurable CPM registry values

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Date	Page	Description of Change
	5-24	Instructions for configuring Front Keyboard updated with details of how to perform this using the Aggregate Builder tool.
	5-25	Instructions for configuring Operator Keyboard updated with details of how to perform this using the Aggregate Builder tool.
	5-27	Added the sequence of message processing following a Suspend Changed online help title from Self Service Support to APTRA XFS; Added note on use of GBRU SP Suspend Timeout
	5-28	Added information on reset command relevant to all Service Providers
	5-33	Added information on updating the position of input data on settlement screens
	5-34	Added cardless transaction registry setting
	5-36	Added note on host control of EJ backup mode; Updated EJ NoOfBackups registry key; Added configuration information for Journal Level options; Added new registry key controlling CD ejection after backup in multiple destinations
	5-37 to 5-40	Added configuration information for the automatic INIT options
	5-40	Added configuration information for EJ compression and maximum file size; Added configuration information for the host control of enhanced EJ mode and EJ checksum
	5-42	Added configuration information for parallel print and encash in BNA Encash, Print and Set Next State; Added configuration information for the alternate journal format for BNA counts
	5-44	Added information on configuring the SP for journalling retract counts; Added information on changing the label when journalling reject counts; Added information on BNA2 cassette mapping for displaying and printing BNA counts through Supervisor;

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Date	Page	Description of Change
	5-45	Added information on choosing the alternative deposit confirmation screen;
	5-45	Updated the GBXX dynamic note sorting information
	5-47	Updated GBXX note reporting information
	5-47 to 5-58	Updated GBXX cassette configuration information
	5-62 to 5-64	Added Barcode Reader configuration information
	5-64	Added mouse pointer information
	5-65	Added diagnostic menu information
	5-66 to 5-68	Added detailed description, installation and configuration information for Silent debug
	5-69 to 5-70	Added detailed description, installation and configuration information for DebugLog
	5-72	Added information on custdat utility
	6-4	Added note on receipt retract functionality
	6-32 to 6-37	Added the contents of the Advance NDC 3.x Test Environment white paper
	6-35	Added section on using VDM with the XFS Simulator
	6-36	Added section on setting up the Simulator for coin dispensers
	7-3 to 7-4	Updated the options for modifying the Customisation Layer
	7-19	Amended field names
	7-21	Added Transaction Status coins and notes dispensed data for field ‘r’
	7-23	Added fields for Transaction Status coins dispensed data when using more than four hoppers and Buffer ‘f’
	10-1	Updated “Overview” with latest changes to this chapter
	10-2 to 10-3	Added instructions for Desktop Installation of the Advance NDC Package

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Date	Page	Description of Change
	10-3	Moved installation information from chapter 6 of the *APTRA Advance ADE, User's Guide*; Added section on the *pmdata* file;
	10-5 to 10-10	Added the content from the Advance NDC Initial Unattended Installation (IUI) and Security white paper
	10-12	Added section on selecting basic or enhanced remote key loading
	10-13	Added information on key mode support for secure and non-secure EPP on NCR and other vendors machines
	10-18	Added note explaining the APTRA Security component cannot be updated.
	10-21	Added post-installation step for updating customised registry keys
	10-22	Updated de-installation information
	Chapter 11	New chapter for troubleshooting information including event and error log information, trace streams, and the use of Silent Debug and DebugLog
	11-9	Moved Problem Determination overview from Chapter 4
	Chapter 12	New chapter on modifying Advance NDC
	A-8	Removed references to CPM and ActiveX from the example of implementing an Authored Exit; Replaced reference to ADE with Author.
	Appendix B	Added description of the following workers: Barcode Reader worker, Barcode Reader Resource ID worker, Extended Cash Stacker, Cheque Acceptor, Coin Dispenser, CPM Initialise, Integer Array Element, NDC Delete RSA Key, NDC Export Encryptor Data, NDC Import RSA Key, NDC Print Coupon, NDC RSA Import DES Key, Resource Status Checker, Resource Initialiser, RSA Key, Variable Graphics Paragraph, XML Config File Loader; Updated the description of the NDC Comms Connection ID
	C-1	Added coin counters CDI store

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Date	Page	Description of Change
	C-3	Added barcode CDI stores
	C-9	Removed BNA Retract table, as these CDI stores are not visible as a catalog in the Author; Added Buffer ‘f’ to Buffer CDI table
	C-11	Added Coin Counters section
	C-12	Added CPM section
	C-28	Added dual dispense CDI stores
	C-30	Added Option Digit CDI stores
	C-41	Updated Default Printer CDI description
	C-47	Added Timer value for Barcode Reader device
	Appendix D	Changed from graphics information (now found in Reference Manual) to font information (moved from Reference Manual)
	Appendix E	Updated the related documentation appendix
	Glossary-2 - Glossary-8	Added definitions for Automatic INIT, Automatic INIT EJ file, coin hopper, coin hopper type, DAPI1, DAPI7, DebugLog, and Silent Debug
May 06		New revision for Advance NDC 3.01
	xxxv	Added information on using and navigating this publication
	xxxviii	Added the two new appendices to the overview
	1-1	Added list of new functionality for release 3.01
	1-1 1-32	Clarified DASH support
	1-20	Removed Enhanced EJ Backup from the list of ignored configuration parameters, as this feature is introduced in 3.01
	1-26	Added GBNA.INI CONFIG menu option to BNA table
	1-30	Corrected article ID for the Microsoft article “Windows Media Player Multimedia File Formats” Added information for dual dispense and web exit enablement and local customisation

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Date	Page	Description of Change
	1-32 to 1-34	Added introductions to the new features introduced in 3.01
	1-35	Removed section on component definition tool as this is no longer supplied
	1-36	Updated information on the APTRA Advance NDC aggregate
	2-2	Updated name of release bulletin file
	2-3	Added information on the User ID utility and updated the installation types information
	2-4	Updated reference to sample application Removed Support Files Directory and replaced with Custom Directory
	3-4 3-6 3-8 3-16 3-17 3-18 3-20 3-21 3-23 A-20 D-1	Updated destination of updated or new files from the *supportfiles* to the *custom* directory.
	3-9	Updated location of changed resrvd.def from supportfiles to the custom directory referenced in the custom.ini file
	4-2	Updated heading to clarify that the interface changes listed are since release 2.x
	4-3	Added option to report individual cassette status for the BNA
	4-5	Added note clarifying that fault display is not supported on BNA devices
	4-8	Added dual cash handlers to the table giving service providers used for cash handler device access
	4-10	Updated BNA reporting option to report on all cassettes, not just the cash bin
	4-14	Added overview of enhanced EJ functionality
	4-16 to 4-18	Added configuration details for dual dispense

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Date	Page	Description of Change
	4-19	Corrected Keep Alive information
	4-20	Updated numeric to alphanumeric dialup information Correct Count Life Time to read Connect Life Time Removed option to rollback the Copy On / Off
	4-22	Added configuration details for setting maximum statement length
	4-28	Changed default to GASPER for SNMP and added SUSPEND_CONDITION trap to NCR SNMP details
	4-29	Updated to say that all cardless settlement screens can be customised
	4-29 to 4-31	Added configuration details for cardless transactions
	4-31 to 4-32	Added configuration details for enhanced EJ backup
	4-32	Added configuration details for dynamic note sorting in GBXX devices Added information on changing the order of the note reporting for GBXX devices
	4-33	Added detailed overview of enabling web exits
	4-34	Added details for specifying local customisation data
	9-3	Removed note on CLM required directory, as CLM no longer supported Removed information on startapps.vbs registering controls as this has been replaced
	9-6 to 9-8	Expanded information on retaining encryption keys
	9-9 to 9-11	Updated the information on preparing a modified application for installation
	F-1	Added new appendix with example *custom.ini* files
	G-1	Added new appendix with configuration reference information for enabling web exits
Dec 05		Updated for Advance NDC 3.00.02
	9-4	Updated registry settings for EPP

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Date	Page	Description of Change
	9-6	Removed *wstart.cmd* as it is no longer used
Sept 05		Updated for Advance NDC 3.00.01
	1-33	Added reference to Internal Multitech Modem MT5634ZPX
	4-23	Added information on Print Track 2 option 37
	4-24	Updated the SP timeout from 60 to 120 seconds
	4-25	Added UPS information and table illustrating power management
	4-26	Added SNMP trap information for heartbeats and mode changes using both NCR SNMP and GASPER-compliant implementations
	4-29	Added information on the customisation of the cardless settlements screen Added information on the AVI file limitation imposed by the operating system
	6-7	Added information on following the author guidelines developed by NCR
Jun. 05		Updated for Advance NDC 3.00.
	All	CEN-XFS 3.00 or later supported; Windows XP supported in the development environment. New functionality: full support for DASH card readers; manual entry of secure keys; new retract and reporting options for the BNA; CCM VISA2 dialup communications; uninterruptible power supply (UPS). Updated tally and error log reporting, as only a defined date and zero values returned; updated file paths as *ulysses* no longer used; updated references to the NCR simulator as only the NCR XFS Simulator is needed. Removed references to Windows NT as no longer supported; removed references to ADI2 and p*ersonaS* SST Device Simulator For Windows NT as no longer supported; BAPE emulation not supported.
	1-15	Updated the data entry mechanism in the Author for the BNA and CPM to refer to new worker, NDC Data Collector.
	1-15	Added comparison of cash dispenser beeper behaviour in NDC+ and Advance NDC.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Date	Page	Description of Change
	1-32	Updated BNA details for retract functionality and additional note reporting.
	1-33	Added paragraphs for CCM VISA2 dialup, DASH readers, secure encryption key entry and UPS.
	3-13	Added details of potential timeout caused by virtual controller processing.
	3-18	In “Differences that Impact Exit Code”, removed “Device Sharing” section as not applicable in CEN-XFS multi-vendor environment.
	4-4	Added overview of Dialup Config menu in Supervisor.
	4-6	Added details of default template file for Problem Determination.
	4-14	Added value ‘000’ to dual mode configuration table.
	4-18	Added paragraph on configuring a spray dispenser to dispense 70 notes.
	4-19 to 4-22	Added section on configuring dialup communications.
	Appx B	Added new workers: NDC Data Collector, NDC Dialup, NDC Echo, Secure Encryption Key Collector, XFS Front FDK, XFS Front Key, and UPS.
	C-12	Added list of CDI stores for dialup.
Oct. 04		Updated for Advance NDC 2.06
	xxxii	Added a reference to the NCR Simulator
	xxxii	Added references to new Chapter 4, “Upgrading from Earlier Releases of Advance NDC” and related details
	xxxiv	Updated course information and added that knowledge of XFS is required for programming to the XFS interface
	xxxv	Updated support information
	1-1	Added information about new functionality in Advance NDC
	1-8 to 1-9	Updated for migration from NDC+ and earlier releases of Advance NDC

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Date	Page	Description of Change
	1-15, 1-29	Added details of support for the BNA and CPM
	1-18	Added paragraph describing the two functions supplied for starting and stopping FDK entry
	1-20	Removed Dual Mode EJ and Hardcopy Backup from list as they are now supported
	1-21	Amended Override Reserved Screens Command differences.
	1-24	Updated Native Mode Option Digit 7
	1-27	Updated communications protocols
	1-28	Added cross-references to publications giving information about using the BNA and CPM
	1-32	Added 'Differences Between Advance NDC 2.06 and Earlier Versions' section
	1-35	Added information about support for RSA initial key loading, support for up to 9999 screens and dual-mode printing
	Chap 2	Updated for: installation of Advance NDC; removal of Windows XP as only Windows NT is supported for a development environment; release bulletin name changed from *AdvanceNDCMMmmpp.txt*
	Chap 3	Added references to the *supportfiles* directory in the Aggregate Builder archive. All references to “system escape” replaced by “exception”
	3-5	Added that format of AVI file names must not exceed 8.3 characters
	3-12	Updated Virtual Controller Restrictions section as no messages can be reflected back to Advance NDC
	3-13, 3-17	Updated description of error handling when loading an Exit from MISCONT or a Supervisor Exit from SUPCTR
	3-24	Amended description of Advance NDC ScreenStore parameters

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Date	Page	Description of Change
	Chap 4	New chapter: “Upgrading from Earlier Releases of Advance NDC”including additional communications configuration for DNS, and PCCM-based protocols
	5-4	Added Journal Configuration section to describe how the journal printer is configured at Start of Day
	5-8, 5-14, 5-18 to 5-22	Updated for new implementation of User Messages and User Terminal Data (was Chapter 4)
	Chap 6	Updated to remove all reference to the Customisation Layer (ADE), which is not provided (was Chapter 5)
	7-1 to 7-14	Updated for new implementation of User Messages and User Terminal Data (was Chapter 6)
	7-20	Added details of placing Supervisor Data Collector last in the work group
	Chap 9	Updated for: installation on the SST desktop; recommendation that Windows XP be used as the operating system on SSTs (was Chapter 8)
	9-1	Added note that not all SPs are started, only those for devices accessed through CEN-XFS
	9-3 9-12	Added information about *startapps.vbs* and how to start Advance NDC on SSTs using the menu shortcut
	A-3	Added recommendation not to use C Exits to extend application functionality
	B-2	Added NDC Print Footer worker
	B-6	Added three workers for RSA
	C-14	Added that binary zeroes are replaced with a question mark in EJ upload
	E-3	Removed reference to *Advance ADE, Seed Application User’s Guide* as it is not supplied with Advance NDC
	E-6	Added reference to *APTRA Communications Feature, User’s Guide*

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Date	Page	Description of Change
	E-7	Added reference to *Extensions for Financial Services (XFS) interface specification (CWA 14050)*
	Glossary	Added CEN, DES, DSM, EPP, HTML, RSA, SP, TTU, VDM and XFS
	Glossary	Updated definitions of User Messages and User Terminal Data

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Contents

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Chapter 2

Installing Advance NDC on a Development PC

Chapter 3

Migrating Existing NDC+ Applications to Advance NDC

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Chapter 4
Upgrading from Earlier Releases of Advance NDC

Chapter 5
Configuring Advance NDC and Support Applications

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and/or distribution is strictly prohibited.

Chapter 6
Introducing the Advance NDC Authored Applications

Chapter 7
Enhancing the Customisation Layer

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Chapter 8

Enhancing the Application Core or Supervisor

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and/or distribution is strictly prohibited.

Chapter 9
Using User-defined CDI Stores

Chapter 10
Delivering an Advance NDC Application to the SST

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Chapter 11

Troubleshooting

Chapter 12

Modifying the Advance NDC Applications

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Appendix A
Using Exits in Advance NDC

Appendix B
Advance NDC Workers Supplied

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Appendix C
Common Data Interface Stores

Appendix D
Font Definition

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

User Feedback Form

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

List of Figures

Chapter 1

Introducing Advance NDC

Chapter 5

Configuring Advance NDC and Support Applications

Chapter 6

Introducing the Advance NDC Authored Applications

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Chapter 7

Enhancing the Customisation Layer

Chapter 8

Enhancing the Application Core or Supervisor

Chapter 10

Delivering an Advance NDC Application to the SST

Chapter 11

Troubleshooting

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and/or distribution is strictly prohibited.

Appendix A

Using Exits in Advance NDC

Appendix D

Font Definition

Appendix F

Example Custom.ini Files

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

List of Tables

Preface

Chapter 1

Introducing Advance NDC

Chapter 3

Migrating Existing NDC+ Applications to Advance NDC

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Chapter 4

Upgrading from Earlier Releases of Advance NDC

Chapter 5

Configuring Advance NDC and Support Applications

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Chapter 6

Introducing the Advance NDC Authored Applications

Chapter 7

Enhancing the Customisation Layer

Chapter 8

Enhancing the Application Core or Supervisor

Chapter 9

Using User-defined CDI Stores

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Chapter 10
Delivering an Advance NDC Application to the SST

Chapter 11
Troubleshooting

Chapter 12
Modifying the Advance NDC Applications

Appendix A
Using Exits in Advance NDC

Appendix C
Common Data Interface Stores

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Appendix D

Font Definition

Appendix E

Related Documentation

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Appendix G

Web Exit State

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

About this Publication

This publication describes the following:

- Upgrading existing APTRA Advance NDC applications
- Migrating existing NCR Direct Connect Plus (NDC+) Native Mode applications to the APTRA Advance NDC product
- Enhancing the APTRA Advance NDC cardholder session, transaction requests/replies and message handling.

For differences in functionality and operation that apply on other vendors' SSTs, refer to the *APTRA Advance NDC, Multi-Vendor Support Reference Manual*.

Using this Publication

This publication can be printed, or viewed electronically.

The electronic publication provides:

- Hyperlinked page references
- A hyperlinked content tree in the Bookmarks pane.

The following navigation and commenting elements are used in this publication:

Table Pref-1
Navigation Elements

Element	Description
Revision Record	Lists updates for supported releases in page order with page references. The most recent release is listed first. Hyperlinked pages are available in the most recent release information only.
Table of Contents	List the first two heading levels in page order with page references.
Table of Contents for chapter or appendix	Lists the first three heading levels in page order with page references.
Glossary	Provides an alphabetical list of terms, acronyms, and abbreviations with their meaning.
Index	Provides an alphabetical list of topics, keywords, concepts, and data references with page references.
Feedback	Allows you to comment on the publication by: — Printing the form — Selecting the Email link — Selecting the Web link.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and/or distribution is strictly prohibited.

Abbreviations used in this Publication

The following abbreviations are used in this publication.

Full Form	Abbreviation
APTRA Self-Service Support	Self-Service Support
APTRA Author	Author
Microsoft Windows NT	Windows NT
Microsoft Windows XP	Windows XP
CEN-XFS 3.00 or later	CEN-XFS 3
HKEY_LOCAL_MACHINE	HKLM
GBNA or GBRU	GBXX

The APTRA Simulator product includes several component parts, including the XFS Simulator and its documentation.

What is in this Publication?

This publication consists of the following chapters and appendices.

Read ...	To ...
Chapter 1, “Introducing Advance NDC”	Familiarise yourself with the Advance NDC components and what documentation can be localised.
Chapter 2, “Installing Advance NDC on a Development PC”	Learn how to install Advance NDC on your development PC.
Chapter 3, “Migrating Existing NDC+ Applications to Advance NDC”	Learn how to migrate an existing NDC+ application to Advance NDC.
Chapter 4, “Upgrading from Earlier Releases of Advance NDC”	Learn how to migrate an existing Advance NDC application and which registry settings you can modify for communications.
Chapter 5, “Configuring Advance NDC and Support Applications”	Familiarise yourself with the Advance NDC Customisation Layer and Application Core.
Chapter 6, “Introducing the Advance NDC Authored Applications”	Familiarise yourself with the Advance NDC application flow in the Author (the Customisation Layer, Application Core and Supervisor applications) and test the applications on a development system

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Read ...	To ...
Chapter 7, “Enhancing the Customisation Layer”	Familiarise yourself with the customisation options available and learn how to modify / enhance the Advance NDC Customisation Layer.
Chapter 8, “Enhancing the Application Core or Supervisor”	Familiarise yourself with the customisation options available and learn how to modify / enhance the Advance NDC Application Core.
Chapter 9, “Using User-defined CDI Stores”	Learn how to create and use User-defined Common Data Interface (UCDI) stores.
Chapter 10, “Delivering an Advance NDC Application to the SST”	Find out what you need to do to install an Advance NDC application on an SST.
Chapter 11, “Troubleshooting”	Familiarise yourself with Advance NDC trace logs and learn how to use the “Silent Debug” and “DebugLog” applications to troubleshoot trace information.
Appendix A, “Using Exits in Advance NDC”	Learn how to implement Exits using different methods.
Appendix B, “Advance NDC Workers Supplied”	Familiarise yourself with the workers provided with Advance NDC.
Appendix C, “Common Data Interface Stores”	Familiarise yourself with the Common Data Interface (CDI) stores and UCDI stores provided with Advance NDC.
Appendix D, “Font Definition”	Familiarise yourself with defining and using fonts
Appendix E, “Related Documentation”	Familiarise yourself with the documentation available to help you migrate to Advance NDC and modify / enhance an Advance NDC application.
Appendix F, “Example Custom.ini Files”	Find out how the *custom.ini* file should look for both Runtime SST and Development environments.
Appendix G, “Web Exit State”	Familiarise yourself with the main parts of implementing a web exit state.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Who Should Read this Publication?

This publication is intended for you if you belong to one of the following categories:

- Current NDC+ Native Mode users who:
  - Want more flexibility than the NDC+ product offers
  - Intend to migrate from OS/2 to an XP environment.
- Users who want the benefits of NDC's message interface, using Self-Service Support. This includes current PAS/NIC and Composer users who wish to utilise the NDC message interface.
- Users who want to run Advance NDC in a multi-vendor environment
- Advance NDC users who want the new functionality offered in this release.
- Advance NDC users who want to move to an SST environment that is compliant with the interface specification for CEN-XFS 3.

What Experience Should I Have?

You are expected to be familiar with using the Windows NT or XP operating systems, and with either NDC+ or Advance NDC. If you are going to develop Exits, a prerequisite is to have work experience of *both* NDC+ and Advance NDC.

For any Advance NDC customisation that requires programming to the XFS interface, such as C Exits or ADE workers, you must be familiar with the CEN-XFS standard and be experienced in programming to the XFS interface.

If considering initial unattended installation, you require knowledge of the following:

- APTRA XFS
- APTRA Security (XP)
- APTRA IUI
- APTRA Aggregate Installer/Builder Tool.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and/or distribution is strictly prohibited.

Training

NCR recommends you complete the courses as described in Table Pref-2.

Table Pref-2
Suggested NCR Courses

Audience	Courses
New to APTRA Advance NDC	*APTRA: History and Structure of NDC;* *Configuring and Testing Windows TCP/IP;* *APTRA Advance NDC - Installing, Configuring and Troubleshooting;* *Preparing APTRA Software for Initial Unattended Installation*
Experience of earlier revisions of Advance NDC	*APTRA Advance NDC 3.2 Update;* *APTRA Advance NDC 3.01 Update;* *APTRA Advance NDC 3.0 Update;* *APTRA Advance NDC 2.6 Update*
Implementing Remote Key Management and / or EMV	*EMV Technical Overview;* *EMV for NDC Overview;* *EMV Solution for NDC Workshop;* *EPP and the Implementation of Remove Key Management (RKM);* *EPP for NDC+ and APTRA Advance NDC*
Enhancing APTRA Advance NDC	*APTRA Advance NDC Implementing Workshop;* *Recommended Practices for Modifying APTRA Advance NDC*
Miscellaneous additional courses	*GBRU/GBNA Configuration;* *Network Management with SNMP;* *SNMP Agent for APTRA, Technical Overview;* *Troubleshooting Using NT Event Logs & APTRA Problem Determination*

The NCR University range of courses is being extended all the time; if you don't currently have a login you can apply for one at *www.ncr.com/ncr_support/ncr_customer_education.jsp*.

Support Information

If you have a problem using Advance NDC, perform the following steps until the problem is fixed:

1 Refer to the *APTRA Author User's Guide*, Appendix A, "Troubleshooting", to see if you can resolve the problem yourself.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Note: If an Advance NDC exception occurs on an SST and Problem Determination Collection has been installed, details are written to the Windows event log to assist in the resolution of the problem.

The *APTRA Author User's Guide* describes how to make use of the information. For example, a domain number is provided, which identifies the offending component. The domain numbers associated with the Advance NDC product are 115 (Customisation Layer) and 116 (Application Core).

2 Contact your internal support person for help.

3 Contact your local NCR representative.

NCR provides a wide range of support programmes. For more information, contact your local Account Team.

If you have any problems using this guide, there is a "User Feedback Form" at the back of this guide, where you will find our email and postal addresses. Please take the time to reply; your comments will be appreciated.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Chapter 1

Introducing Advance NDC

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and/or distribution is strictly prohibited.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Overview

This release of Advance NDC offers the following:

- A migration path from NCR Direct Connect Plus (NDC+) Native Mode to the NCR APTRA Author Environment (Author)

  Note: The NDC+ Diebold Emulation product cannot be migrated to Advance NDC. For more information, refer to the *APTRA Advance NDC, Overview*.

- A multi-vendor environment supporting the same functionality as Advance NDC 2.06, except for tally reporting and error log reporting.

This chapter discusses the following:

- The Advance NDC applications
- Differences between Advance NDC and NDC+
- Differences between Advance NDC 3.0x and earlier versions
- The Aggregate Builder Tool (ABT) and Advance NDC.

For information about running Advance NDC on other vendors' SSTs, refer to the *APTRA Advance NDC, Multi-Vendor Support Reference Manual.*

For information about new features in Advance NDC, refer to the *APTRA Advance NDC, Overview*.

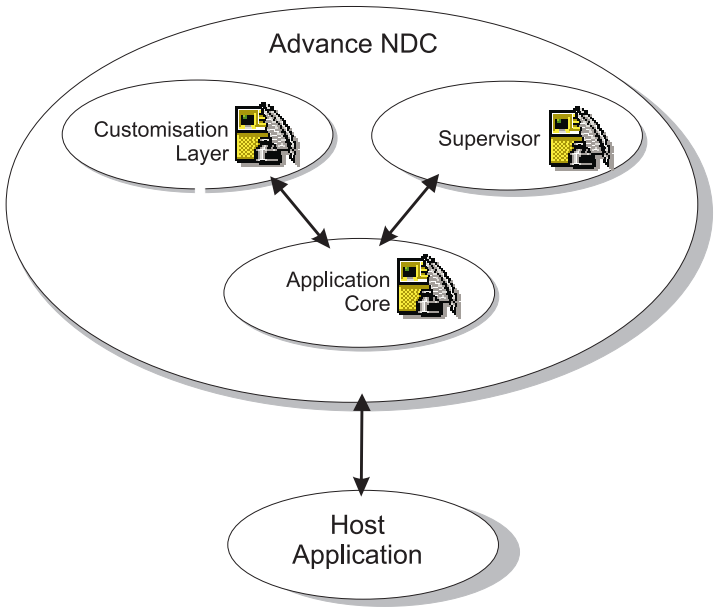
Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Advance NDC Applications

Prior to Advance NDC 2.05, the Supervisor functionality was contained in the Application Core. In Advance NDC 2.05, the Supervisor functionality was moved from the Application Core into a separate Supervisor application, giving Advance NDC three applications developed using the APTRA Author, NCR's graphical development tool.

The three applications are the Supervisor, Application Core and Customisation Layer, and they interact with each other and the Central application as shown in the following diagram:

Figure 1-1
Advance NDC Applications



When the applications are running, control passes between them at several key points. The Central application resides on a host or switch and interacts with Advance NDC on the SST to perform self-service transactions and maintain the SST in operation.

The Application Core performs the background processing activities previously performed by the executable part of NDC+. The Supervisor performs the 'Out Of Service' activities associated with the supervisor. The Customisation Layer performs the 'In Service' activities associated with the cardholder.

Advance NDC worker classes are used in these applications to implement NDC-specific features.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Shared data is used by the Application Core, Supervisor and Customisation Layer. The repository that holds this data is called the Common Data Interface (CDI).

The following sections describe each of these applications.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Application Core

The Application Core is an authored application performing the tasks of mode handling and message handling.

The Application Core does not execute States and Screens—this is done solely by the Customisation Layer.

Mode Handling

The mode handling portion of the Application Core is responsible for moving from one mode to another, based on certain conditions of the Self-Service Terminal (SST), as well as performing the functionality of each mode.

The modes are:

- Out of Service
- Supervisor—*see* **Note:**
- In Service
- Offline.

Note: Entry to and exit from Supervisor Mode is handled by the Supervisor application. As in NDC+, Supervisor Mode is entered only when the terminal is idle and after the mode switch is moved to the 'Supervisor' position. Supervisor Mode functionality is also provided by the Supervisor application; *see* "Supervisor" on page 1-6.

The Supervisor application uses the Common Data Interface (CDI) to read and set configuration options and counters. For details of the CDI, *see* the "Common Data Interface" on page 1-10.

In addition to the modes, there are other tasks that are performed by the Application Core:

- Power-up
- Start of Day
- Initialise
- Suspend.

Message Handling

As well as mode handling, the Application Core also handles the messages from Central. It either accepts or rejects the incoming messages depending on the content of the message and the current mode of the SST. This shields the Customisation Layer from any message handling and retains the integrity of the NDC message interface compatibility. Any messages received which are not valid for a particular mode are rejected, in exactly the same way as in NDC+.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

As in NDC+, Central can send the following types of messages to the SST:

- Terminal Commands
- Customisation Data Commands
- Transaction Reply
- Host To Exit.

For details of these messages, refer to the *APTRA Advance NDC, Reference Manual*.

Enhancing the Application Core

The Application Core executes the fixed part of NDC+ that would not typically be customised. Most developers will only wish to make changes to the Customisation Layer but, like NDC+, the Application Core can be customised using the following:

- Screen downloads
- Configuration and / or Enhanced Configuration Parameter downloads
- Native Mode option digits
- Reserved Screens.

The Application Core cannot be modified to the same extent as the Customisation Layer; but you can extend the NDC Message Interface to process a new Message Class or add additional data to Terminal State messages. You enhance the Application Core using the APTRA Author, and Authoring components (Workers).

Any modifications to the Application Core must be made by observing the customisation guidelines given in Chapter 7, “Enhancing the Customisation Layer”. Conformance to the guidelines will ensure the core NDC+ functionality (mode handling and existing messages) remains intact and that minimal redevelopment is required for subsequent releases of Advance NDC.

Any Application Core customisations you make using the Author will have to be reapplied to future releases of Advance NDC Service Packs or updates. For this reason, NCR recommends you avoid using the Author to customise the Application Core.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Supervisor

The Supervisor is an authored application that handles Supervisor Mode entry / exit, and performs the tasks associated with the supervisor.

The Supervisor interface has the same look and feel as in NDC+. However, you can customise the Supervisor functionality in the following ways:

- Modifying the text displayed in menus and messages
- Modifying the text used for printing Security Trace Messages
- Including your own Supervisor Exits.

It is possible to add new menus and functions using the APTRA Author. However, Supervisor customisations you make using the Author will have to be re-applied to future releases of Advance NDC or service pack updates. For this reason, NCR recommends you avoid using the Author to customise the Supervisor application.

For more information on how the Supervisor application handles Supervisor Mode, see “Supervisor Mode” on page 6-14.

To enable data to be shared between the Application Core and the Supervisor application, some Common Data Interface (CDI) stores are shared stores. These stores are listed in “Shared Stores” on page 6-31.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Customisation Layer

The Customisation Layer is an authored application that handles all aspects of a cardholder session (for example, Card Entry, Card Read, PIN Entry, Transaction Request/Reply, Cash Dispense). In other words, it performs the 'In Service' activities associated with the cardholder.

The equivalent functionality of the Customisation Layer in NDC+ is when NDC+ enters In Service Mode and executes the downloaded State Flow, starting at State 0 (the Card Read or Card Read PIN Entry Initiation State).

The Customisation Layer allows you to customise the behaviour of the SST to a greater extent than was available in NDC+. The aim of the Customisation Layer is to allow you to customise the behaviour of the cardholder session without impacting the Application Core or the NDC message interface.

The Customisation Layer runs concurrently with, and is driven by, the Application Core. Any processing performed by the Customisation Layer can only be done while the SST is In Service.

The Application Core controls the Customisation Layer by communicating mode changes to and from In Service. However, if the SST is In Service and in the middle of a cardholder session, the Customisation Layer will complete the current session before passing control back to the Application Core.

For details of the method used to synchronise the Customisation Layer and Application Core, *see* "Synchronisation Between the Customisation Layer, Application Core and Supervisor" on page 1-9.

Migrating from NDC+ to Advance NDC

In Advance NDC, NCR provides a Customisation Layer, which executes your existing NDC+ download. There is no need to modify the download in any way.

Chapter 3, "Migrating Existing NDC+ Applications to Advance NDC", describes how you can use Advance NDC to execute your existing NDC+ download. However, NCR recommends you first read this chapter and Chapter 2, "Installing Advance NDC on a Development PC".

Enhancing the Customisation Layer

After migrating to Advance NDC, you may want to offer, during In Service mode, functionality that cannot be achieved using NDC+ State Types. However, any customisations you make will have to be

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and/or distribution is strictly prohibited.

re-applied to future service packs for and new releases of Advance NDC.

In NDC+ you were able to customise the behaviour of the SST using data download messages. Advance NDC supports this method of customisation and additionally offers a more flexible customisation approach using the Author.

You can enhance the Customisation Layer provided or develop your own, using the APTRA Author and Authoring components (workers).

By modifying the Customisation Layer using the Author, you can author new cardholder services and transactions, and incorporate them into an existing NDC+ State Flow with no impact to the rest of the application. Modifying the Customisation Layer is discussed in "Methods of Enhancing the Customisation Layer" on page 7-4

Any modifications you make to the Customisation Layer using the Author will have to be reapplied to future releases of Advance NDC or service pack updates.

Upgrading from an Earlier Release of Advance NDC

Upgrading from an earlier release of Advance NDC may require some additional configuration as all devices are now accessed through an interface that is compliant with the CEN-XFS 3 specification, instead of ADI2 or CEN-XFS 2.

For details of the changes affecting customisation and configuration in Advance NDC, see Chapter 4, "Upgrading from Earlier Releases of Advance NDC".

If you are upgrading to move to a multi-vendor environment, the *APTRA Advance NDC, Multi-Vendor Support Reference Manual* gives details of the operational and other differences that apply on other vendors' SSTs.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Synchronisation Between the Customisation Layer, Application Core and Supervisor

The main role of the Customisation Layer is to initiate a cardholder session which performs cardholder transactions. In order for this processing to occur, the SST must be in the In Service mode and ready to initiate a cardholder session. The Application Core synchronises with the Customisation Layer allowing a cardholder session to be initiated.

While the SST is In Service, the Application Core can query the status of the Customisation Layer using a shared store provided by the Customisation Layer. This is used to determine whether further message processing or mode changes can take place at various stages in the application execution (such as processing messages while the SST is idle, waiting for a card to be entered).

The method used to synchronise the Customisation Layer and the Application Core is known as the Session Request/Release protocol. This method is described in the “Wait For In Service” on page 6-5 and “Synchronising with the Customisation Layer” on page 6-16.

Note: The synchronisation of Supervisor entry/exit and the Supervisor functionality is handled by the Supervisor application.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and/or distribution is strictly prohibited.

Common Data Interface

The Application Core, Customisation Layer and Supervisor share data at runtime. The repositories that hold this data are called the Common Data Interface (CDI) and the User-defined Common Data Interface (UCDI).

Common Data Interface

The CDI consists of stores representing all of the data used throughout an Advance NDC application. There are several hundred CDI store workers, one for each piece of data.

The CDI stores are provided in "CDI - <name>" catalogs. For details of the CDI stores provided and their use, see Appendix C, "Common Data Interface Stores".

Each piece of CDI data has a specific purpose and is used in different parts of the Advance NDC application. For example, the Operation Code Buffer is read by the Transaction Handler and included in the Transaction Request Message. It is written to by various State Types, and may be written to by your Customisation Layer.

The workers are grouped in catalogs according to their use. Examples of CDI data are as follows:

- Configuration Options
- FIT Data
- Transaction related buffers, such as the Operation Code Buffer, Buffer A and Card Track 3 Data Buffer.

The CDI stores you are most likely to access are those in the "CDI - Buffers" catalog.

General Purpose Buffers

Many of the buffers used in Advance NDC store a particular data item, such as the Card Track 1 Data buffer for the Card Read State, or the Amount buffer for the Amount Entry State. There are also two general purpose buffers, Buffer B and Buffer C, which can be used to store a variety of numeric data collected from the cardholder. The functionality of the buffers is determined by the download from Central, or a local customisation.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Different state types can populate these buffers or be used to check the values stored in them, as follows:

Table 1-1
Use of the General Purpose Buffers

State Type	Description
H - Information Entry State	Used for data of up to 32 digits. Examples of the data that could be stored are: an account number for a money transfer; or a mobile phone number when buying more call time.
G - Amount Check State	Used to check the value of the buffers against specified limits.
R - Enhanced Amount Entry State	Used when multi-language screens are required; the specified buffer stores the amount entered by the cardholder. The maximum number of digits that can be entered is determined by the size of the Amount buffer, 8 or 12 digits.
X - FDK Information Entry State	The specified buffer stores a predefined number corresponding to the FDK selected by the cardholder. For example, this could be used to offer fixed amounts of cash to the cardholder.

The locally stored data can be passed to Central, which must specify transmission of the buffers, using a Transaction Request state if required. Central also specifies which buffer to use.

In addition, Buffer B is used in an Interactive Transaction Response, when a message is sent to the SST with screen data and specific keys enabled. The key selected by the cardholder at this screen is reported to the host in a second transaction request message containing the single keypress value stored in Buffer B. If the cardholder presses the Cancel key, or does not press a key before a timeout occurs, then the value C or T is returned in the buffer to indicate the cardholder's action.

For details of the state types, Transaction Requests and Interactive Transaction Responses, refer to the *Advance NDC Reference Manual*.

Migrating from NDC+

If you are simply using Advance NDC to execute your existing NDC+ download, you do not need to concern yourself with the Common Data Interface as access to the CDI stores has been implemented for you.

All of the data in Advance NDC will be accessed at the correct time and place, whether it is from one of the NDC+ State Types or from a Transaction Request/Reply.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and/or distribution is strictly prohibited.

Enhancing the Customisation Layer

Whether you choose to enhance the Customisation Layer or develop your own, you need to be aware of the relevance of the Common Data Interface and the use of the CDI workers we provide.

The CDI has the following two roles:

Table 1-2
CDI Roles

Type of CDI Data	Role
Data used by the Advance NDC application	Enables data sharing between State Types, the Transaction Request / Reply mechanism and workers.
Data that was made available in NDC+ to allow you to develop Exits (Shared Data)	Enables Exit code you have developed to migrate to Advance NDC.

The data that you need to access is dependent on what you want to do. NCR has provided all of the data you will need.

If you are developing a Customisation Layer that does not make use of NDC+ State Types, you need to write to the correct CDI workers before using the Transaction Request / Reply mechanism. For example, you need to explicitly put a value in the Operation Code Buffer, which is included in the Transaction Request message. To do this, you would write to the Operation Code Buffer worker in the "CDI - Buffers" catalog, using a Setter / Assigner worker.

For details of the Setter and Assigner workers, refer to the worker class on-line help provided with the APTRA Author.

For details of the CDI stores you need to write to for each of the Transaction Request Fields, see "Transaction Request / Reply" on page 7-16.

For more details of NDC+ shared data, refer to the *Using NDC Exits* publication.

User-defined Common Data Interface

The APTRAUSERCDI local service runs at start-up. It is identical to the CDI in functionality, but also enables you to create new CDI stores (called UCDI stores) without the need to modify the CDI C++ source code.

Like CDI stores, UCDI stores provide a means of sharing data between applications. They hold string or integer data and may be persistent (persistent data is preserved across terminal shutdowns / power failures).

You can create UCDI stores to share data between the Customisation Layer and Application Core. For example, you may

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

use UCDI stores when supporting a third party device. You would also use UCDI stores when configuring new options or timers.

You must ensure all UCDI stores are cleared, initialised and set by your application, as required.

An example of a UCDI store implementation is the CPM Cheque Accept State ('w'). The state uses a UCDI store named CPM cheque present which checks if a cheque is present in the CPM device. The store is used in the state and by Supervisor.

For details of the UCDI stores provided and their use, see Appendix C, "Common Data Interface Stores".

For more information on using UCDI stores, see Chapter 8, "Enhancing the Application Core or Supervisor".

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Differences Between Advance NDC and NDC+

To compare the overall features offered by NDC+ with those offered by Advance NDC, refer to the *APTRA Advance NDC, Overview*.

This section discusses specific operational differences between Advance NDC and NDC+. It focuses on the features in NDC+ that are either not available in Advance NDC or operate differently. It also includes features in Advance NDC that either extend or were not present in NDC+. These operational differences include:

- Aggregate Building
- Cash dispenser beeper
- Cash dispenser cassette types
- TM-Alert
- Document Processing Module (DPM)
- Coin Dispenser
- Digital Audio Service (DAS)
- State Types
- Configuration Parameters
- Customisation Data Commands
- PIN Entry and Verification
- BAPE emulation
- Electronic Journal and Journal Printer Backup
- Screen Data
- K Screens
- Diebold Emulation Mode Status Messages
- Status Messages
- Option Digits
- Supervisor Mode
- Differences in MACing
- Communications Protocols
- Basic Operator Panel (BOP)
- VGA Enhanced Rear Operator Panel (VEROP)
- Security Camera
- Software Management
- Associated Keyboards
- Bunch Note Acceptor (BNA)
- Cheque Processing Module (CPM)

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

- Transaction Request Extension State
- Font Definition
- Character Sets
- Localising Advance NDC
- Unsolicited Status Messages
- MPEG File Support
- Maximum State Number
- Number of Screens Supported
- Dual Cash Handlers
- Web Exit Enablement and Customisation.

BNA and CPM

The BNA and CPM sections are also included for your convenience. Note that these are not strictly NDC+ differences, but Advance NDC additional features.

Keyboard entry for the BNA makes use of external C functions to enable and disable the FDK keyboard, including FDK touch emulation.

If you intend customising the Advance NDC application, you must use the NDC Data Collector worker for touch screen and keyboard data entry.

Aggregate Building

From Advance NDC 2.05, the Aggregate Builder Tool has been used to define component aggregates.

For an introduction to building aggregates with Advance NDC, *see* "Aggregate Building" on page 1-42.

Cash Dispenser Beeping

In NDC+, when option A was set to Action (do not retract notes), option B was ignored; so the beeper did not sound. In Advance NDC, the behaviour of option B, when set to Action, depends on the setting of option A. For details, see the *APTRA Advance NDC, Supervisor's Guide*.

Cash Dispenser Cassette Types

In NDC+, up to four cassette types were supported.

In Advance NDC, Enhanced Configuration option 76 controls the number of supported cassette types. By default, this option is set to support up to four cassette type. For further details of option 76, refer to the *APTRA Advance NDC, Reference Manual.*

TM-Alert

TM-Alert is not supported by Advance NDC.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

The following table details which message types associated with TM-Alert are handled differently by Advance NDC.

Table 1-3
TM Alert

Message Class	Message Sub-Class	Message	Destination	Differences
5	-	TM-Alert	Terminal	This message from the Central application is not supported. If it is sent, Advance NDC responds with a Command Reject or Specific Command Reject (E01).
3	-	TM-Alert	Central	As a result of the above, Advance NDC does not send this message to the Central application.

Document Processing Module

The Document Processing Module (DPM) is not supported in Advance NDC.

Advance NDC behaves in the same way as NDC+ when running on an SST where a DPM device is not present.

The following table details the message types associated with this device which are handled differently by Advance NDC.

Table 1-4
DPM Message Types in Advance NDC

Message Class	Message Sub-Class	Message	Destination	Differences
1	1	Transaction Request	Central	If the DPM fields are requested for inclusion in this message, they will be included. Although all DPM data (such as Read Zones) is initialised, it is never written to by Advance NDC. NCR recommends you do not request inclusion of the DPM related fields.
1 2	2 2	Device Fault Status Information	Central	Advance NDC will not send any Unsolicited or Solicited Device Fault Status messages for the DPM.
1	2	Sensors Device Fault Status Information	Central	For the Sensors device, byte 19 of the Device Status is not reported as it relates to the DPM.
2	2	Send Configuration Information Terminal State Message	Central	DPM, and DPM Tamper Indicator (TI) and Coin TI fields are included in this message, if specified, but will always indicate that the device is not configured.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Message Class	Message Sub-Class	Message	Destination	Differences
2	2	Send Supply Counters Terminal State	Central	DPM fields are not included in this message.
2	2	Send Tally Information Terminal State	Central	Not supported in Advance NDC. A default message is always returned in which zeros are reported for the DPM.
2	2	Send Error Log Information Terminal State	Central	Not supported in Advance NDC. DPM fields are not included in the SST Device Error Logs.
2	2	Hardware Configuration Data Terminal State	Central	There will be no Device Identifier / Configuration for the DPM in this message (Device Identifier ‘]’). If a Depository is present, Device Identifier ‘F’ will have a Hardware Configuration of ‘02’ for PPD (Programmable Printing Depository) reported rather than ‘04’ for DP-ATM Depository. If a Statement printer is present, Device Identifier ‘V’ will have a Hardware Configuration of ‘01’ for standard Statement printer reported rather than ‘05’ for 5665 Statement printer. If an Envelope Dispenser is present, Device Identifier ‘\’ will have a Hardware Configuration of ‘01’ reported rather than ‘02’ for DP-ATM Envelope Dispenser. There will be no Device Identifier / Configuration for DPM TI (‘_’).
2	2	Supplies Data Terminal State	Central	There will be no Device Identifier / Supplies for DPM. DPM TI is never reported.
2	2	Fitness Data Terminal State	Central	There will be no Device Identifier / Fitness for the DPM. DPM TI is never reported.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Message Class	Message Sub-Class	Message	Destination	Differences
2	2	Tamper and Sensor Status Data Terminal State	Central	In field g2, the value of the sensor status is derived from the XFS HRESULT. In field g3 (Tamper Indicator Identifier 'B') the status of each Tamper Indicator is reported. This field is always 13 bytes in length. Bytes 1-7 are derived from XFS. Bytes 8-12 deal with the coin dispenser. Byte 13 is always '0', indicating DPM TI is not present. In field g4 (Extended Tamper Indicator Identifier 'C') the status of each extended Tamper Indicator is reported.
4	-	Transaction Reply	Terminal	Functions 'S' and 'T' are not supported by Advance NDC. If they are specified, a Command Reject or Specific Command Reject (E02) is reported. These functions are rejected at the earliest stage, with no validation or use of the DPM data in the Transaction Reply. Printer Flags ';' and '<' are ignored by Advance NDC.
3	1	Override Reserved Screens Command	Terminal	This Customisation Data Command (Message Identifier 'G') is accepted, but the 'Type D' reserved screens are not used by Advance NDC. A Ready 9 message is returned.
3	1	Configuration / Enhanced Configuration Parameters Load	Terminal	These Customisation Data Commands (Message Identifiers '3' and 'A') are accepted. *See* "Configuration Parameters" on page 1-23.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

The following table details which State Types associated with the DPM device are handled differently by Advance NDC.

Table 1-5
DPM State Types in Advance NDC

State Type	Description
Close	The extension to this State related to DPM error handling and screens is ignored.
Courtesy Amount Verification	This State does nothing other than take the exit specified by Table Entry 4 (CAV Unavailable Next State Exit).
DPM Document Accept	This State does nothing other than take the exit specified by Table Entry 6 in Extension 3 (Exception Type 3 Next State Exit).
Enhanced Amount Entry	The “Start CAV” extension to this State is ignored by Advance NDC.

Coin Dispenser

The differences between using a coin dispenser with NDC+ and using a coin dispenser with Advance NDC are detailed in Table 1-6.

Table 1-6
Coin Dispenser Differences

NDC+	Advance NDC
Supported a coin dispenser with four hopper types	By default, Advance NDC is NDC+ compatible and configured to support only four hopper types. However, support for up to eight hopper types can be set using Enhanced Configuration option 79
Supported a coin dispenser with a maximum dispense of 25 coins	Supports the dispense of coins up to the maximum reported by the hardware SP
Coins can be dispensed using the Display and Print function	Coins cannot be dispensed using the Display and Print function
Coins cannot be dispensed without notes	Coins can be dispensed without notes; a cash dispenser does not have to be present for a coin dispense
A low condition is reported immediately, then the transaction continues	A low condition is reported at the end of the transaction
The STD COIN Supervisor option prints all counters before performing the CLR COIN, ADD COIN and CHECK CDM Supervisor options	The STD COIN Supervisor option prints the coin counters before performing the CLR COIN, ADD COIN, and CHECK COIN Supervisor options

If no coin dispenser is configured, Advance NDC returns the same message as NDC+ when running on an SST where a coin dispenser is not present.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

The following table details which message types associated with this device are handled differently by Advance NDC.

Table 1-7
Coin Message Types in Advance NDC

Message Class	Message Sub-Class	Message	Destination	Differences
1	1	Transaction Request	Central	If option 79 is set to 000, the message has the same format as in NDC+ with field 'r' containing the number of coins dispensed in the last transaction. If option 79 is set to 001, the number of coins dispensed is reported in the 'ce1' to 'ce*<n+1>*' fields (derived from buffer f) and field 'r' is set to zero.
1 2	2 2	Device Fault Status Information	Central	If option 79 is set to 000, device fault status information is sent for four hopper types. If option 79 is set to 001, device fault status information is sent for the number of hopper types reported in the Hardware Configuration message. A minimum of four hopper types are reported. The Advance NDC message also reports hopper type severities and the actual coins dispensed.
2	2	Send Configuration Information Terminal State Message	Central	If specified, a Tamper and Sensor Status Data message is returned for the coin dispenser. The information is extended, as shown in Tamper and Sensor Status Data Terminal State below.
2	2	Send Supply Counters Terminal State	Central	Using the command modifiers for Command Code '4', it is possible to select the basic or extended message format. The basic message format is as NDC+ (command modifier 1). The extended message format differs from NDC+. It allows the reporting of more than four hopper types (command modifier 2).
2	2	Send Tally Information Terminal State	Central	This is not supported in Advance NDC. For further information, refer to the *Advance NDC, Reference Manual*.
2	2	Send Error Log Information Terminal State	Central	This is not supported in Advance NDC. For further information, refer to the *Advance NDC, Reference Manual*.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Message Class	Message Sub-Class	Message	Destination	Differences
2	2	Hardware Configuration Data Terminal State	Central	If enhanced configuration option 79 is set to 001, the coin dispenser configuration data is extended from 2 to 7 bytes. This provides additional information on the number of hopper types and the maximum coins per dispense.
2	2	Supplies Data Terminal State	Central	Extended to allow the reporting of more than four hopper types.
2	2	Fitness Data Terminal State	Central	Extended to allow the reporting of more than four hopper types.
2	2	Tamper and Sensor Status Data Terminal State	Central	In field g2, the value of the sensor status is derived from the XFS HRESULT. In field g3 (Tamper Indicator Identifier 'B') the status of each Tamper Indicator is reported. This field is always 13 bytes in length. Bytes 1-7 are derived from XFS. Bytes 8-12 deal with the coin dispenser. Byte 13 is always '0', indicating DPM TI is not present. In field g4 (Extended Tamper Indicator Identifier 'C') the status of each extended Tamper Indicator is reported.
4	-	Transaction Reply	Terminal	Extra fields for more than four hopper types are used if option 79 is set to 001. Printer Flags ';' and '<' are ignored by Advance NDC.
3	1	Override Reserved Screens Command	Terminal	This Customisation Data Command (Message Identifier 'G') is accepted, but the 'Type D' reserved screens are not used by Advance NDC. A Ready 9 message is returned.
3	1	Configuration / Enhanced Configuration Parameters Load	Terminal	These Customisation Data Commands (Message Identifiers '3' and 'A') are accepted. *See* the "Configuration Parameters" section.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Digital Audio Service

The features added to NDC+ release 6 relating to the Digital Audio Service (DAS) are not supported in Advance NDC.

The following table details which message types associated with the DAS are handled differently by Advance NDC.

Table 1-8
DAS Message Types in Advance NDC

Message Class	Message Sub-Class	Message	Destination	Differences
4	-	Transaction Reply	Terminal	Printer Flag '>' (Annunciated Voice) is ignored by Advance NDC.
2	2	Terminal State	Central	The DAS device is not reported in any Terminal State messages.
1 2	2 2	Device Fault Status Information	Central	Advance NDC will not send any Unsolicited or Solicited Device Fault Status messages for the DAS.
3	1	Enhanced Configuration Parameters Load	Terminal	This Customisation Data Command (Message Identifier 'A') is accepted, however the Enable Audible Echo of Keyboard parameter and the option to report DAS errors to the Central application are ignored. *See* "Configuration Parameters" on page 1-23.

The Audio Control State does nothing other than take the Next State Number specified by Table Entry 4 (Function Complete Next State Number).

Note: Although the DAS is not supported by Advance NDC, industry standard Sound Blaster WAVE (*.wav*) and MIDI (*.mid*) audio files can be used. For details, see "Recreating Audio Files" on page 3-5.

State Types

Advance NDC handles the following State Types differently from NDC+:

- Close - see "Document Processing Module" on page 1-16
- Courtesy Amount Verification - see "Document Processing Module"
- DPM Document Accept - see "Document Processing Module"
- Enhanced Amount Entry - see "Document Processing Module"
- Audio Control State - see "Digital Audio Service" on page 1-22
- PIN Entry - see "PIN Entry States" on page 1-24
- Enhanced PIN Entry - see "PIN Entry States"
- Card Read - PIN Entry Initiation - see "PIN Entry States"
- Camera Control - see "Security Camera" on page 1-31

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

- Smart FIT check - refer to the *APTRA Advance NDC, Reference Manual*, Chapter 12, 'Smart Card Handling'

Configuration Parameters

The following Configuration and / or Enhanced Configuration Parameters may be downloaded, but are ignored by Advance NDC:

- Camera Control (Security Camera is not supported)
- Card Read Error Threshold
- Card Write Error Threshold
- Timers:
  - 70 and 71 (Not supported in Advance NDC)
  - 82 and 83 (Security Camera is not supported)
  - 91 (In Service EJ Log Inspection is not supported)
  - 92 (SOH reporting is not supported)
  - 97 (Door Access is not supported)
- Diebold Status Reporting for Vandal Guard (Diebold is not supported)
- Tamper Indication Control Option
- Extended Status Control Option
- Optical Sensor Option (CIM86 is not supported)
- Journal Printer Backup Log Tamper Option
- Touch Screen Error Reporting Option (Touch Screen errors are not reported because, unlike S4, Self-Service Support does not support a proprietary Touch Screen device)
- Remote Relay Option (Remote Relay is not supported)
- TPA Informed of SM Activity Option (not applicable with Self-Service Support)
- Include PAN in DCS Data
- Enable Audible Echo of Keyboard (DAS is not supported)
- Report DAS Errors (DAS is not supported)
- Non-Magnetic Card Accept.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Customisation Data Commands

Advance NDC handles Customisation Data Commands differently from NDC+, as listed in the following table:

Table 1-9
Customisation Data Commands in Advance NDC

Message Class	Message Sub-Class	Message	Destination	Differences
3	1	Diebold PIN Information Load	Terminal	This Customisation Data Command (Message Identifier '4') is accepted, however the downloaded data is not stored because it is never used in Advance NDC. A Ready 9 message is returned.
3	1	Override Reserved Screens Command	Terminal	This Customisation Data Command (Message Identifier 'G') is accepted; the 'Type D' reserved screens are used by Advance NDC for EMV Exits.
3	1	Configuration / Enhanced Configuration Parameters Load	Terminal	These Customisation Data Commands (Message Identifiers '3' and 'A') are accepted. For details of unsupported options, *see* the "Configuration Parameters" on page 1-23.

PIN Entry and Verification

The following PIN Verification methods are not supported by Advance NDC:

- Local Diebold PIN Verification
- Local GBP PIN Verification
- Atalla PIN Verification
- Remote BANKSYS PIN Verification.

PIN Entry States

In Advance NDC, the numeric keypad must be used for PIN entry. As a result of this, the following State Types do not support the use of the Touch Screen in Advance NDC:

- PIN Entry
- Enhanced PIN Entry
- Card Read - PIN Entry Initiation.

PIN Block - No Encryption

A value of 4 for the FIT field PINPD means no Encryption of the PIN Block. This value is not supported in Advance NDC as it breaches the use of Secure PIN Entry and results in the PIN being sent to the Central application in clear.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

In Advance NDC, use of Secure PIN Entry has been made, which means the PIN the cardholder enters is not available to the application.

Specifying a value of 4 for PINPD will result in the Cancel Next State being taken in PIN Entry States.

BAPE Emulation

The EPP in BAPE emulation mode is not supported in Advance NDC.

Electronic Journal and Journal Printer Backup

The following journal-related features are not supported, or are supported differently in Advance NDC.

Electronic Journal Log File Checksum

This feature is supported differently in Advance NDC.

When the Electronic Journal (EJ) was initialised (using the INIT EJRNL Supervisor function), NDC+ added an eight-byte ASCII numeric string (which acted as a checksum) to the end of both the file copied to diskette (*ejxxxxxx.nnn*) and the backup log file (*ejrcpy.log*). This checksum was also logged at the start of the new journal log file (*ejdata.log*).

In Advance NDC, this checksum feature has also been implemented as part of the Electronic Journal functionality. However, the checksum can be turned off using a registry setting. For details of turning off the checksum, see "EJ Checksum" on page 5-41.

Note: Adding a checksum results in a longer initialisation process that increases with larger EJ files. For details of the INIT EJRNL option, refer to the *APTRA Advance NDC, Supervisor's Guide*.

EJ Log File Inspection (In Service Supervisor)

This feature is not supported in Advance NDC.

If the EJ was configured and the SST had a rear operator panel, NDC+ supported inspection of the EJ log file and the copying of sections of the log file to diskette (as DOS text) while the SST was in any Mode. Additionally, copying sections of the log file to a printer was supported while the SST was in Supervisor Mode.

Tampered Journal Records

This feature is not supported in Advance NDC.

If Journal Printer Backup was configured, NDC+ applied an integrity check to all records passing through the Journal Printer Backup disk buffer. If an integrity error occurred on a reprint record, NDC+ supported several different courses of action (all

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

configured by the Journal Printer Backup Log Tamper Enhanced Configuration Parameter) as follows:

a No action

b Printed the defective ('tampered') record, bracketed by >> and << characters

c Sent an unsolicited status message, including the last Security Trace Number, its date and time stamp, and the record offset, to the Central application

d In addition to option **c**, wrote the defective journal record to a file on the system disk (up to ten defective records).

In the case of option **d**, subsequent access to these files used the FREE JDATA Supervisor function, enabled or disabled by a Terminal Command from Central.

This 'tamper' feature has not been implemented as part of the Journal Printer Backup functionality in Advance NDC, as follows:

- No checks are made for 'tampered' reprint records
- No unsolicited status messages relating to this feature are sent to the Central application
- The FREE JDATA Supervisor function has not been implemented. However, if the Enable Free JData Terminal Command is received from the Central application, it will be handled by Advance NDC in the same manner as by NDC+. Setting the associated Enhanced Configuration Parameter (22) will have no effect on the system, as it will be ignored by Advance NDC.

EJ File Format

This feature is supported differently in Advance NDC.

The EJ file contains the same printable characters as were logged using NDC+. However, there may be additional Advance NDC specific printer control codes present in the file.

In NDC+, the EJ file size was fixed at 1.44 MB. Advance NDC, allows a file size to be set for the EJ of at least 1K. The maximum that can be set is calculated using the available disk space. For details, see "Maximum EJ File Size" on page 5-41.

Advance NDC also does the following to the EJ log file:

- Adds an end date stamp
- Allows password protection and compression
- Allows compression without password protection.

If you have developed a tool to view the Electronic Journal file, ensure that it ignores irrelevant control characters.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Screen Data

The following Screen Data features are not supported by Advance NDC:

- Text and cursor blinking - ignored
- Set display mode control - ignored, because the screen resolution cannot be changed at runtime
- Voice - Antex Audio files are not supported. Instead industry-standard Sound Blaster WAVE (*.wav*) and MIDI (*.mid*) audio files can be used. For details, see "Recreating Audio Files" on page 3-5.
- Animation - VGM*.ani* files and Autodesk FLIC (*.fli* / *c*) animation files are not supported. Instead, Microsoft AVI (*.avi*) movies and industry standard MPEG (*.mpg*) video / audio files can be used. For details, see "Recreating Animation Files" on page 3-6.

The following screen resolutions are not supported:

- 640 x 350 x 16
- 320 x 200 x 256.

Advance NDC supports resolutions of 640 x 480 and above. For details, see "Recreating Graphics" on page 3-3.

K Screens

In NDC+, screens K04, K05, and K06 were sent to the printers on exit from NDC+ to prepare for diagnostics.

In Advance NDC, these screens are not used. For details, refer to the *APTRA Advance NDC, Reference Manual.*

Diebold Emulation Mode Status Messages

No Diebold Emulation Mode Status messages are sent to the Central application because Advance NDC only supports NCR Native Mode.

Status Messages

Advance NDC does not send Unsolicited / Solicited Device Fault Status information messages for the following devices:

- Touch Screen Keyboard - unlike S4, Self-Service Support does not support a proprietary Touch Screen device
- Document Processing Module - device not supported. See "Document Processing Module" on page 1-16
- Digital Audio Service - device not supported. See "Digital Audio Service" on page 1-22
- Security Camera - device not supported. See "Security Camera" on page 1-31
- Door Access - device not supported.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Table 1-10
Status Message

Message Class	Message Sub-Class	Message	Destination	Differences
1 2	2 2	Card Reader / Writer Device Fault Status Information	Central	The Transaction / Device Status values of '6'-'9' are not reported, because the CIM86 Card Reader is not supported. Similarly, the Error Severity is always one byte.
1	2	Sensors Device Fault Status Information	Central	*See* "Document Processing Module" on page 1-16.

Option Digits

The following Option Digits (specified in Supervisor mode) can be entered but are ignored:

- Option Digit 1
- Option Digit 3, Part B
- Option Digit 6, Part A
- Option Digit 7, Part B; Part C - SOH reporting is handled differently by Self-Service Support compared to NDC+, refer to the *APTRA Advance NDC, Supervisor's Guide* for details.

Supervisor Mode

Advance NDC does not use Reserved Screens M10, M11 and M13 to define key positions on the Cardholder Keyboard. Instead, Advance NDC Supervisor mode uses the default keyboard layouts provided in NDC+.

If you wish to change the key positions and return code, you can do so using the TTU SP configuration for rear operator access or the PIN SP configuration for front operator access.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

The following Supervisor functions are not supported by Advance NDC:

Table 1-11
Supervisor Functions Not Supported

Menu	Options	Present in the Advance NDC Supervisor Menu?
Select Menu	SET SW0 - SET SW3 CSOH 2 CSOH 3 TM-ALERT	NO
Replenish Menu	FILL CAMERA DUMP IMAGES	YES
Configure Menu	VOLUME SST VOLUME JACK	YES
Access Menu	INIT SUP SOH INIT INS SOH FREE JDATA	NO

For details of the behaviour in Advance NDC of the unsupported functions, refer to the *APTRA Advance NDC, Supervisor's Guide.*

Communications

Advance NDC provides Supervisor menus for configuring CCM VISA2 dialup or TCP/IP communications with Central. These menus are accessed by selecting the required menu function on the Configure Menu.

If you wish to use CCM VISA2 dialup, or TCP/IP to communicate with the host, refer to the *APTRA Advance NDC, Supervisor's Guide* for details of the TCP/IP and dialup configuration menus. For additional communications configuration, *see* "Configuring Communications" on page 5-11.

Bunch Note Acceptor (BNA)

Advance NDC has the following menu options for the BNA, if it is present:

Table 1-12
BNA Menu Options

Menu	Options	Function
Configure	DISP CSH/ACC CONFIG	Display the BNA configuration
	PRNT CSH/ACC CONFIG	Print the BNA configuration
	GBNA.INI CONFIG	Configure the *GBNA.ini* file
	GBRU/GBNA CONFIGURATION	Configure the GBXX device

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and/or distribution is strictly prohibited.

Menu	Options	Function
Replenish	INIT BNA	Initialise the BNA
	CLR BNA	Clear the BNA counts
	DISP CNTRS	Display the BNA counters
	PRNT CNTRS	Print the BNA counters

Refer to the *APTRA Advance NDC, Supervisor's Guide* for details of the Replenish and Configure menu functions for the BNA.

Diagnostics

If you move the mode switch to Normal, then exit Diagnostics in Advance NDC, the SST returns to In Service or Out of Service mode.

In NDC+, this returned you to Supervisor mode.

Miscellaneous Functions

Advance NDC provides a Select menu item for Miscellaneous Functions ('MISC FUNCS'). The MISC FUNCS menu contains options for journalling the EMV smart card application component versions ('PRNT CMPNT VERS') and smart card hardware version ('PRNT SCRW VERS'), if present.

Differences in MACing

If the security flags are such that flag 1 is FALSE and flag 2 is TRUE, a Message Authentication Code (MAC) value is expected at the end of the following messages, but it is not checked:

- Transaction Reply
- Customisation Data State Tables Load Command
- Customisation Data FIT Data Load Command
- Customisation Data MAC Field Selection Load Command
- Dispenser Currency Cassette Mapping.

When security flag 1 is set to FALSE (do not check MAC in incoming messages), no attempt is made to determine whether or not a MAC exists in the message. When security flag 2 is set to TRUE, this indicates that a MAC is expected. Do ***not*** use this combination of security flags because the consequences are as follows:

- Some invalid messages may be rejected with a different reject status from before. This is not seen as an issue, as invalid messages should not be sent to a live SST.
- Messages that are valid, but do not have a MAC attached when one is expected, will be accepted. Previously such messages would have been rejected.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Communications Protocols

APTRA Communications Connection Manager (CCM) is used in Advance NDC to support the following protocols for communications:

- TCP / IP
- VISAII
- All protocols supported by the PCCM service except for IBM 3600 Loop communications.

Basic Operator Panel (BOP)

The Basic Operator Panel (BOP) is not supported by Advance NDC. Although the Supervisor interface has the same look and feel as in NDC+, menus can only be displayed on the front interface or the Enhanced Operator Panel (EOP).

VGA Enhanced Rear Operator Panel (VEROP)

In NDC+, the VGA Enhanced Operator Panel (VEROP) could be used in EOP emulation mode. However, Self-Service Support does not support the VEROP in EOP emulation mode. For Advance NDC, this means that if you wish to use an operator panel, it must be the Enhanced Operator Panel (EOP).

Security Camera

The Integrated Security Camera device is not supported in Advance NDC.

Advance NDC behaves the same way as NDC+ running on an SST where the Security Camera device is not present.

The following table details which message types associated with the Security Camera device are handled differently by Advance NDC.

Table 1-13
Security Camera Message Types in Advance NDC

Message Class	Message Sub-Class	Message	Destination	Differences
1	2	Device Fault Status Information	Central	Advance NDC will not send any Unsolicited Device Fault Status messages for the Security Camera.
2	2	Send Configuration Information Terminal State Message	Central	Security Camera fields are included in this message, if specified, but will always indicate that the device is not configured.
2	2	Send Supply Counters Terminal State	Central	The Camera Film Remaining is always returned as zeros.
2	2	Send Tally Information Terminal State	Central	Zeros are reported for the Security Camera in this message.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Message Class	Message Sub-Class	Message	Destination	Differences
2	2	Send Error Log Information Terminal State	Central	Security Camera fields are not included in the SST Device Error Logs.
2	2	Hardware Configuration Data Terminal State	Central	There will be no Device Identifier / Configuration for the Security Camera in this message (Device Identifier 'M').
2	2	Supplies Data Terminal State	Central	There will be no Device Identifier / Supplies for the Security Camera in this message.
2	2	Fitness Data Terminal State	Central	There will be no Device Identifier / Fitness for the Security Camera.

The Camera Control State does nothing other than take the next State number specified by Table Entry 3 (Next State Number).

Software Management

In NDC+, a message was sent to the Central application (Message Class 4, Sub-Class 1) giving information on the status of Software Management installation.

In Advance NDC, there is no interaction between Advance NDC and the Software Management components running in the SST. Therefore, Advance NDC will never send Message Class 4 to Central and the Enhanced Configuration Parameter 'TPA Informed of SM Activity Option' is redundant.

Associated Keyboards

In NDC+, a screen displayed during a State Type could optionally have had an associated keyboard definition, to let you define which keys to enable and the return code for each key.

In Advance NDC, these associated keyboards are ignored, as keyboard layouts are now defined at the PIN SP and platform level.

Bunch Note Acceptor (BNA)

The Bunch Note Acceptor (BNA), if present, is supported in Advance NDC as follows:

- Additional information in various Terminal State Messages
- Terminal Command - Retrieve Note Definitions
- Device Fault Status messages
- Configuration Parameters
- State Type - Cash Accept State
- Additional fields in Transaction Request and Reply messages
- Three transactions - two Encash functions and a Refund transaction
- Changes to various Supervisor replenishment functions

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

- Specific Supervisor functions
- Changes to Start of Day
- CDI Stores.

For details of using the BNA, refer to the following publications:

- *APTRA Advance NDC, Supervisor's Guide*
- *APTRA Advance NDC, Reference Manual.*

Cheque Processing Module (CPM)

The Cheque Processing Module (CPM), if present, is supported in Advance NDC as follows:

- Additional information in various Terminal State Messages
- New Device Fault Status messages
- New Configuration Parameters
- New State Type - Cheque Accept State
- Additional fields in Transaction Request and Reply messages
- New Print Flags allowing cheque endorsement and / or rubber stamping
- Changes to Close State
- Changes to various Supervisor replenishment functions
- Two Supervisor functions
- Returns cheques not authorised by Central
- Endorses and pockets authorised cheques
- Changes to Start of Day
- New CDI Stores.

For details of using the CPM, refer to the following publications:

- *APTRA Advance NDC, Supervisor's Guide*
- *APTRA Advance NDC, Reference Manual.*

Transaction Request Extension State

The Transaction Request Extension State has been modified to allow you to add new data fields to the Transaction Request message. For more details, see "Adding Data Fields to a Transaction Request" on page 7-16.

Font Definition

With Advance NDC, you can define alternative fonts to those provided with Advance NDC. These alternative fonts can be third-party fonts, or fonts you have developed.

For more information, see Appendix D, "Font Definition".

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Character Sets

The following NDC+ designated character sets are not supported in Advance NDC:

- '6' (Customer Graphics 1)
- '8' (Customer Graphics 3)
- '9' (Customer Graphics 4)
- '<' (Chinese 3)
- '=' (Chinese 4)
- '@' (Chinese 7)
- 'A' (Chinese 8).

Localising Advance NDC

To aid your understanding of the Advance NDC Customisation Layer and Application Core, and the NDC-specific workers provided with Advance NDC, on-line information in UK English is available in the form of On-line Help and Component Descriptions.

You can translate the on-line information if required. To learn how to do this, refer to the *APTRA Advance ADE, Local Language Customisation Guide.*

For details of the files which contain the on-line information to be translated, refer to the text files *<global>\Custom\CustomisationLayer\rt115_cust.txt* and *<global>\Custom\ApplicationCore\rt116_cust.txt*.

Unsolicited Status Messages

You may find that some unsolicited status messages, which were not sent in NDC+, are sent to Central in Advance NDC. For example, with Advance NDC, if the currency dispenser shutter is jammed closed when a present is attempted, a Suspend unsolicited status message is sent to Central. With NDC+, this Suspend message was not sent.

With Advance NDC, unsolicited status messages may also be sent at different times as compared with NDC+.

MPEG File Support

NDC+ supported MPEG files through the hardware, but Advance NDC uses Microsoft's MCI interface (Video for Windows) to render video files (AVIs, MPEGs). On Windows XP, Windows Media Player uses the DirectX interface for video rendering. For details of formats supported on Windows XP, refer to the Microsoft article "Windows Media Player Multimedia File Formats" (article ID 316992) at http://support.microsoft.com.

Maximum State Number

In Advance NDC 2.05, the maximum number of valid states was increased from 750 to 999. Each 'Next State Number' entry in the *APTRA Advance NDC, Reference Manual,* Chapter 2 "State Tables" had a possible range of 000-254 or 256-999.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

From Advance NDC 3.02, the maximum number of valid states is increased from 999 to 46655. This is achieved by introducing support for alphanumeric state numbers in state tables. Depending on the number system specified in Enhanced Configuration Parameter option 80, each 'Next State Number' entry in the *APTRA Advance NDC, Reference Manual*, Chapter 2 "State Tables" has a possible range as follows:

- 000-254 or 256-999 using the decimal (base 10) system
- 000-254 or 256-ZZZ using the alphanumeric (base 36) system.

Number of Screens Supported

NDC+ supported up to 999 screens. From Advance NDC 2.06, up to 9999 screens are supported.

Dual Cash Handlers

Dual cash handlers, if present, are supported in Advance NDC as follows:

- Increased cash handler capacity without host impact
- The ability to use each cash handler for a different currency type
- Two cash handlers are reported as a single logical cash unit
- The option of using up to seven cassette types, depending on the setting of Option 76
- Separated and / or combined status reporting for hardware configuration, supplies and fitness data.

Web Exit Enablement

From Advance NDC 3.01, web exits are supported.

Web exits allow the display of screens and interaction with the consumer to occur outside of Advance NDC.

Providing Local Customisation Data

From Advance NDC 3.01, local customisation data can be used to replace or modify that provided in the host downloads.

The local data is provided by files placed directly on the SST.

Support for Screens

The following differences exist between NDC+ screens and Advance NDC screens:

- X screens — In NDC+, X screens were used to display state of health information. In Advance NDC, X screens are used for audio messages
- C05 screen — In NDC+, if a card jam occurred resulting in a suspend condition, screen C05 and another 'card could not be read' screen were displayed. In Advance NDC, only screen C05 is displayed in this situation.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Set Display Mode Control Sequence

In NDC+, the Set Display Mode control sequence could be used.

Advance NDC does not support this control sequence.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Differences Between Advance NDC 3.x and Earlier Versions

This section summarises the changes and additions in Advance NDC 3.x.

New Feature Support

The following new devices or new functions are supported:

- BNA additional functionality
- CCM VISA2 dialup communications
- Full support for DASH card readers for EMV
- Secure key entry at the SST
- Uninterruptible Power Supply (UPS)
- Dual cash handlers
- Cardless transactions
- Alphanumeric dialup
- Enhanced EJ features
- Multiple destinations for the EJ
- Display and print TCP/IP configuration and advanced options
- Configuring the minimum and maximum statement length.
- Enabling web exits and local customisation.
- Silent Debug trace capture on live SSTs
- DebugLog trace viewing in the development environment
- Support for reflection of SST to Host messages.

BNA

BNA devices are supported through ActiveXFS.

BNA 2 devices also offer the following :

- Options to switch retract functionality on and report retract operations
- Configuration of additional reporting options.

GBXX GBRU devices are supported for deposit or dispense. However, recycling is not supported.

From release 3.01 the following functionality is supported for GBXX devices:

- Cassette status can be optionally reported in the host message when option digit 45 is set to enable the extended message format

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and/or distribution is strictly prohibited.

- Dynamic note sorting can be configured when a cassette becomes full. This allows the device to continue accepting notes and to store them in sequenced cassettes. This is optional, and configured using registry settings.
- If a GBNA device is detected, a *GBNA.ini* file is created. This file allows the selection of denominations accepted by the device, and the ordering of denominations in the note report to host. The denominations can also be selected (activated) or deselected (deactivated) through the Supervisor configuration menu. The order of reporting is set by the order of the denominations in the *GBNA.ini* file.

Dialup communications

Advance NDC and the Internal Multitech Modem MT5634ZPX use CCM VISA2 for dialup communications. There is an additional menu in Supervisor to configure most of the dialup parameters. In the APTRA Author, there is a new worker class, NDC Dialup, and a CDI store, Dialup Flag, to handle dialup communications in the Supervisor project.

DASH Card Readers

In the APTRA Author, a new worker class, Card ATR Reader, and a work group has been added to the Card Acceptor worker class, to provide smart card support for EMV.

Secure Key Entry

Manual entry of secure keys is supported with a new worker class, Secure Encryption Key Collector.

Note: Secure key entry offers standards compliance rather than enhanced security. The decision to use secure key entry rather than entering keys through the EOP is the responsibility of the financial institution.

Uninterruptible Power Supply (UPS)

Support for UPS devices enables the SST to complete a transaction if a power failure occurs, and go out of service, or perform a controlled shutdown if the battery power runs low.

Dual Cash Handlers

From release 3.01, APTRA Advance NDC supports the capability to increase cash handler capacity without host impact.

A second cash handler can now be added with the two cash handlers reported as a single logical cash unit, therefore emulating a single cash handler.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Cardless Transactions

From Advance NDC 3.01, a non-cash transaction can be started without a card using FDK, FDK emulation, or a numeric key defined in the registry. This provides an alternative to using a card.

Alphanumeric Dialup

Alphanumeric characters can now be included in the following dial up fields:

- BIN
- TID
- MODEM INI STR
- PRIMARY NO
- SECONDARY NO
- N/A STRING

Enhanced EJ Features

Various EJ options have been added since release 3.01.

EJ Backup Mode In previous releases of Advance NDC, only the latest copy of the EJ could be retained. From release 3.01, up to 1000 previous backups can be retained, depending on disk space. The option is enabled through the Enhanced Configuration Parameters download, or using the Supervisor configuration menu. The format of the EJ remains unchanged.

Host Control of BackUp Mode From release 3.02, the Enhanced Configuration Parameters download method of updating the EJ backup mode can be blocked. This option is enabled through the Supervisor configuration menu.

Multiple Destinations for EJ Backup In addition to retaining multiple copies of the EJ, the destination for the EJ backup file can be selected from available network and removable drives such as CDRW, diskette or USB (Universal Serial Bus) drives. This option is enabled through the Supervisor configuration menu.

Maximum File Size From release 3.02, a maximum size can be set for the EJ file. This can be a minimum of 1KB, the maximum depends on available disk space. This option is enabled through the Supervisor configuration menu.

Automatic INIT From release 3.02, several methods are available to allow EJ INIT to be run automatically, as follows:

- Cutover
- Scheduled
- Agent

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and/or distribution is strictly prohibited.

There is also an option to take a copy of the INIT file after completion of the INIT operation. All these options are enabled through the Supervisor configuration menu.

Direct EJ Backup From release 3.02, an emergency backup can be made of the EJ file if, for example, there is insufficient disk space for an INIT operation. This is selected through the Supervisor replenishment menu.

Compression of the EJ File From release 3.02, the EJ file can be compressed as part of the INIT operation.

Privacy for the EJ File From release 3.02, the EJ file can be compressed and password protected as part of the INIT operation.

Display and Print TCP/IP Configuration

From release 3.01, options are available through Supervisor to display or print the current TCP/IP configuration information. If the print option is selected, the default printer for the operator interface is used (receipt printer for the facia interface and journal printer for the operator interface).

Configure Statement Length

From release 3.01 a maximum statement length for the statement printer can be specified using a registry key. This allows the setting of the maximum number of lines to be printed before a cut. This can be used, for example, where black mark paper is not used and no form feed is included in the transaction reply print data.

Enable Web Exits

From release 3.01, Advance NDC provides support for enabling web exits. Using web exits provides a means to interact with the consumer outside of Advance NDC.

Specify Local Customisation Data

Local customisation allows the configuration download from the host to be overridden or modified by files placed on the SST

Silent Debug

From release 3.02, Advance NDC provides support for a utility called Silent Debug that will perform trace capture on live SSTs.

DebugLog

From release 3.02, Advance NDC provides support for the DebugLog utility.

Note: The DebugLog utility was available in previous releases of Advance NDC but was not supported.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and/or distribution is strictly prohibited.

Message Reflection

From release 3.02, Advance NDC supports reflection of terminal to host messages using the virtual controller interface. Previously, only host to terminal message reflection was supported. Reflected messages are recorded in the `MESSAGEOUT` and `MESSAGEIN` trace streams. Reflected messages in these trace streams can be viewed using the Silent Debug or DebugLog utilities.

For information on viewing trace streams using Silent Debug or DebugLog, *see* "Troubleshooting Tools" on page 11-9.

Device Access

Prior to Advance NDC 2.06, only ADI2 was used for device access. Advance NDC 2.06 introduced, for some devices, an interface compliant with the CEN-XFS 2 specification. In Advance NDC 3.00 all devices, with the exception of CPM, were accessed through an interface compliant with CEN-XFS 3. From Advance NDC 3.02 all devices are accessed through an interface compliant with CEN-XFS 3.

For details of how this affects customised applications, see Chapter 4, "Upgrading from Earlier Releases of Advance NDC".

Support for Screens

From Advance NDC 2.06 onwards, up to 9999 screens are supported. Prior to Advance 2.06, Advance NDC supported up to 999 screens.

Remote Key Management

From Advance NDC 2.06, for SSTs fitted with an Encrypting PIN Pad (EPP), the loading of the initial encryption keys (A key, B key and V key) using RSA encryption is supported as an alternative to local entry in Supervisor mode.

Dual-Mode Journal Printing

From Advance NDC 2.06, additional functionality on the Supervisor Configure menu supports dual-mode journalling, which allows both a paper journal to be printed and an electronic journal (EJ) to be created.

Unsupported Features

The EPP in BAPE emulation mode is no longer supported in Advance NDC.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Aggregate Building

Using the ABT (Aggregate Builder Tool), you have control over the components to be installed on the development PC or the SST. For information about the ABT, refer to the on-line help provided with the tool.

Advance NDC Aggregate

From Advance NDC 3.01, there is only one Advance NDC installable aggregate. The aggregate provides the following installation options:

- Development
- Development and Simulation
- Runtime SST
- Documentation Only.

For details of the components in the Advance NDC Package, refer to the *APTRA Advance NDC, Overview*.

Customising Aggregates

If you wish, you can customise the provided aggregate installation by using the Aggregate Builder Tool (ABT) on the development PC. For details, see “Preparing a Modified Advance NDC Aggregate for Installation” on page 10-18.

You can also use the Aggregate Builder Tool to control the installation of components on a development PC, using a similar process.

User-Created Components

Using the Component Builder, you can build your own components (for example, a custom presentation client) to add to an aggregate.

For guidance, refer to the Component Builder on-line help.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Chapter 2

Installing Advance NDC on a Development PC

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Overview

The Advance NDC package includes the Advance NDC product together with other associated products. See Chapter 10, “Delivering an Advance NDC Application to the SST” for details of the following:

- A secure initial unattended installation of Advance NDC on an SST
- Desktop installation of Advance NDC on an SST
- Advanced (custom) installation of Advance NDC

This chapter provides additional information about installing Advance NDC on your development PC and what to do after installation.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

System Requirements

For details of the hardware required and prerequisite software for your development PC, refer to the *APTRA Advance NDC Overview* (PDF) which is available in the APTRA online documentation after Advance NDC has been installed, or in the root directory of the CD-ROM before installation.

For last-minute information refer to the release bulletin (*ANDC.mht*).

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Installation Process

Before installing the Advance NDC software, ensure your development PC meets the requirements specified in the *APTRA Advance NDC Overview* (PDF) which is available in the APTRA online documentation or in the root directory of the CD-ROM.

Before You Start

If the development PC already has Advance NDC or the APTRA Author installed, you must remove them as described in "De-installing Advance NDC" on page 10-22.

If you select **Change Folder**, you will be prompted to enter the location and name of the Advance NDC global directory. This is where Advance NDC will be installed. If you do not specify a global directory, the default name of *<drive>:\ntglobal* will be used. NCR recommends you obtain the details before you start.

Before using the Author for the first time, you must also set up at least one User ID. This can be completed as the last step of installation, or at a later time, using the User ID utility. For details of this utility and reserved User IDs, refer to the *APTRA Author User's Guide*.

Installation Types

From release 3.01, all installation options are available for selection from a single Advance NDC aggregate.

For the development PC, the Development and Simulation option provides the same functionality as the Development option, and also allows you to run simulations of your developments.

If you want to see the documentation before installing the Advance NDC software, you can select the Documentation Only option in the Target Environment window during installation. This will install all the documentation for Advance NDC and other components in the Advance NDC aggregate. You do not need to de-install the documentation before installing the software.

For detailed information about installing Advance NDC on an SST, see Chapter 10, "Delivering an Advance NDC Application to the SST".

For more information about the components that are installed on the development PC or SST, refer to the *APTRA Advance NDC Overview* (PDF), which is available in the APTRA online documentation or in the root directory of the CD-ROM.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Installing Advance NDC on a Development PC

Installing Advance NDC on your development PC for the *first* time is like installing any software package on the Windows XP operating system.

Caution: Any previous installation of prior to Advance NDC 3.01 or the APTRA Author must be removed first. For details, refer to "De-installing Advance NDC" on page 10-22.

To install Advance NDC, perform the following steps:

1 Insert the APTRA Advance NDC Package CD-ROM into CD-ROM drive of the development PC.

2 Choose to do one of the following:

- Run the *setup.exe* file located in the root directory and follow the on-screen instructions for the installation type you require
- Read the *APTRA Advance NDC Overview* (PDF), which is available in the APTRA online documentation or in the root directory of the CD-ROM
- Read the release bulletin.

3 After installation is complete, read and follow the instructions in "After Installation" on page 2-4.

You can also modify the software to create an aggregate using the Aggregate Builder Tool. For more information, see Chapter 10, "Delivering an Advance NDC Application to the SST".

For detailed information on installing the Silent Debug utility, see "Silent Debug Installation" on page 5-66.

For detailed information on installing the DebugLog tool, see "DebugLog Installation" on page 5-69.

APTRA Author Sample Application

A sample application can be obtained for the APTRA Author by contacting NCR Dundee. For details, refer to the *APTRA Author, User's Guide.*

After Installation

Once you have successfully installed Advance NDC on your development PC, you need to set up required files and folders to use the Aggregate Builder Tool and / or the APTRA Author for any modifications you intend making to the Advance NDC application supplied in the Advance NDC Package.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Custom Directory

From release 3.01, the custom directory replaces the Advance NDC Support component containing the *supportfiles* subdirectory.

This directory is where you place your DLLs and any other modified files. The custom directory can be created anywhere within the APTRA component directory. The ABT recognises any folder containing *_comp.ini* as an APTRA component directory. The *custom.ini* file is used to locate user files and include them in the aggregate.

For details, see "Preparing a Modified Advance NDC Aggregate for Installation" on page 10-18.

Working Directory for the APTRA Author

To use the APTRA Author, you need to create a working directory to store your projects and generate a final build. For details, refer to the *APTRA Author, User's Guide*.

For information about running the Advance NDC application on your development PC, see Chapter 6, "Introducing the Advance NDC Authored Applications".

De-installing Advance NDC

For details of the files to back up and how to de-install Advance NDC and related components, refer to the release bulletin on the Advance NDC CD-ROM.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Chapter 3

Migrating Existing NDC+ Applications to Advance NDC

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Overview

This chapter guides you through the migration of an existing NDC+ application to Advance NDC.

This chapter provides information on:

- How to migrate an NDC+ download, including the creation of:
  - Graphics files
  - Audio files
  - Animation files
  - Reserved screens
  - Communications Template files
- Migrating existing NDC+ Exits
- Proving the download works.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Executing an Entire NDC+ Download

Migrating an existing NDC+ application to Advance NDC requires little effort. The Customisation Layer supplied by NCR enables you to use your existing NDC+ download in a Windows XP environment, unchanged.

The Customisation Layer handles the download and execution of NDC+ customisation data (States Types, Screen Data, Printer Data, Configuration Parameters and Financial Institution Tables). **There is no need for you to make any authoring changes using the APTRA Author**.

Note: If you want to familiarise yourself with the application flow, you can do so using the Author. See Chapter 6, “Introducing the Advance NDC Authored Applications”. If you intend enhancing the Customisation Layer, see Chapter 7, “Enhancing the Customisation Layer”. To enhance the Application Core, see Chapter 8, “Enhancing the Application Core or Supervisor”.

Changes Required

To successfully run your Customisation Layer application, you may need to recreate or migrate certain files, as described in the following table, which contains links to the relevant sections in this chapter:

Table 3-1
Help for Recreating or Migrating Files

To...	See Section...
Recreate graphics files	“Recreating Graphics” on page 3-3
Recreate audio files using new tools	“Recreating Audio Files” on page 3-5
Recreate animation files using new tools	“Recreating Animation Files” on page 3-6
Recreate communications template files	“Recreating Communications Template Files” on page 3-9
Migrate Exits used in your NDC+ application	“Migrating Existing NDC+ Exits” on page 3-10
Edit *resrvd.def* if you have made changes to it	“Changing RESRVD.DEF” on page 3-8

After carrying out the relevant changes, you can proceed to “Proving the Download Works” on page 3-25.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Recreating Graphics

Advance NDC does not support the 640 x 350 and 320 x 200 screen resolutions used in NDC+. Advance NDC only supports screen resolutions of 640 x 480 and above.

If you have created your own graphics for display in NDC+, you may need to recreate them. If you try to display one of your current NDC+ graphics using Advance NDC, the vertical dimension of the graphic, as displayed on the screen, will be reduced by approximately 28%.

After recreating your own graphics, copy them to your custom directory. This includes them in the aggregate when you export it from the Aggregate Builder Tool.

Although you need to recreate your graphics, download screens do not require any changes, as the same screen controls as in NDC+ are supported (or ignored if not supported; for example, the 'Set Display Mode Control' is ignored if sent).

Note: Download screens containing Display Image Files Controls may reference files using a pathname; any path should be removed.

The *config.con* file is no longer required. In the following subsections, we detail the names of files required to successfully display logos and pictures in Advance NDC.

Cardholder Graphics

To take account of the different screen resolutions supported, the Screen Type 'G' cardholder graphics (G00, G01 and G03-G06) supplied with NDC+ have been recreated. For more information, refer to the *APTRA Advance NDC, Reference Manual.*

The provided graphics are located in the *<global> \ final \ xfs \ dll* directory.

Copy the required graphics to your custom directory. This includes them in the aggregate when you export it from the Aggregate Builder Tool for installation on an SST.

You do not need to make any changes to your screen definitions to use these graphics.

Note: The supplied picture files *pic**xxx**.pcx* are used instead of the NDC+ *G**xx**.pcx*. Do not change or remove any of these picture files.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Logo Control

Logos must have a file name of the format *logo**xx**.**yyy***, where ***xx*** is the two digit number representing the logo (for example, 00) and ***yyy*** is its file extension.

For details of the supported graphics file formats, search for *Picture File Formats* in the APTRA Author online help.

Picture Control

Pictures must have a file name of the format *pic**xxx**.**yyy***, where ***xxx*** is the three digit number representing the picture (for example, 000) and ***yyy*** is its file extension.

Although in NDC+ pictures could be referenced by a number of between one and three digits, Advance NDC requires that the file name has three digits. Leading zeros are expected in the file name (for example, *pic**020**.yyy*). You do not need to modify the screen definitions. A screen definition of `ESC 'P' 2 20 ESC '\'` will display the file *pic**020**.yyy*.

Note: The supplied picture files *pic**xxx**.pcx* are used instead of the NDC+ *G**xx**.pcx*, and should not be changed or removed. The *G**xx**.pcx* pictures should also not be moved or deleted.

Refer to the *APTRA Advance, GUI Help* for details of the supported graphics file formats.

Display Image Files Control

When referencing a file from a Display Image Files Control, do not include the path. An AVI file-name prefix must not exceed eight characters, that is, the name must conform to the 8.3 format (<*filename*>.*avi)*.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Recreating Audio Files

Advance NDC does not support NCR's proprietary Antex audio board. Therefore, Antex audio files cannot be played in Advance NDC.

Instead industry standard Sound Blaster WAVE (*.wav*) and MIDI (*.mid*) audio files are supported and played via the Media Control Interface (MCI).

The same host (Central) interface for playing audio is supported. As in NDC+, a Screen control results in an audio file being played.

The following table identifies how Advance NDC handles playing audio files compared with NDC+.

Table 3-2
Audio File Handling in Advance NDC and NDC+

	NDC+	Advance NDC
Number of standard messages provided	10 Antex audio files	10 WAVE / MIDI audio files (same messages as in NDC+)
Screen control plays a message identified by a number	'xx' or 'xxx'	'xx' or 'xxx'
Audio file format (where 'xx' or 'xxx' maps the file to a message number)	*daudio**xx**.msg* or *daudix**xx**.msg*	*daudio**xx.yyy*** or *daudi**xxx.yyy*** (see step 1 on the next page)

To play message number 'xx' or 'xxx', you need to do the following:

1 Re-record or save the audio file as a *.wav* or *.mid* file.

Note: The file must follow the same fixed naming convention used in NDC+. That is, *daudio**xx.yyy*** or *daudi**xxx.yyy***, where ***xx*** or ***xxx*** is the message number and ***yyy*** is *wav* / *mid*.

2 Copy the file to your custom directory.

By following the file format, file name convention and directory location specified, the audio files will be included in the aggregate when you export it from the Aggregate Builder Tool. For information about using the Aggregate Builder Tool with Advance NDC, refer to the *APTRA Author User's Guide.*

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Recreating Animation Files

Advance NDC does not support proprietary VGM *.ani* animation files, or Autodesk FLIC (*.fli* or *.flc*) animations.

Instead, Microsoft AVI (*.avi*) movies and industry standard MPEG (*.mpg*) video / audio files are supported.

Note: There is a system-imposed limit on the number of AVI files that can be used. Refer to "AVI System Limit" on page 5-33 for further information.

The same host (Central) interface for playing animations is supported. As in NDC+, a Picture control or Display Image Files control can be used to display an animation file.

The following table identifies how Advance NDC handles displaying animation and video files compared with NDC+.

Table 3-3
Video File Handling in Advance NDC and NDC+

	NDC+	Advance NDC
Control displays a picture (animation) identified by a number	'x'	'x'
Mapping between picture (animation / video) file and number is specified by (Picture control only)	*config.con*	A fixed mapping convention is used *pic**xxx**.**yyy*** (see step 1 on the next page)

To display an animation or video file with number 'xxx', complete the following:

1 Create the animation or video as an *.avi* or *.mpg* file.

Note: 1. To display using a Picture Control sequence, Advance NDC requires that an animation file name has three digits; any leading zeros are required in the file name. That is, *pic**xxx**.**yyy***, where ***xxx*** is the picture (animation / video) number and ***yyy*** is *avi* or *mpg*. For example, *pic**020**.avi*.

Note: 2. If the animation is played using the Display Image Files control, no mapping is required. When referencing a file from a Display Image Files Control, do not include the path.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

2 Copy the file to your custom directory.

By following the file format, file name convention and directory location specified, the animation or video files will be included in the aggregate when you export it from the Aggregate Builder Tool. For information about using the Aggregate Builder Tool with Advance NDC, refer to the *APTRA Author User's Guide*.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Changing RESRVD.DEF

If you have made any changes to the *resrvd.def* file in NDC+, you will need to re-apply these changes for Advance NDC.

Note: The *resrvd.def* file must not contain any TAB characters. Use the SPACE character instead.

To apply changes without editing the supplied *resrvd.def* file, do the following:

1. Create a file that uses the same format as the *resrvd.def* file, but which contains your changes

2. Update the following registry key with the name of your file:

```
HKLM\SOFTWARE\NCR\Advance
NDC\supervisor\UserAdditionalReservedScreenFile
```

Valid values are blank for no additional reserved screen file, or the name of the file, for example *resrvd.4012*.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Recreating Communications Template Files

When an SST running NDC+ software is installed in a network, its High Order communications have to be configured before it can communicate with the Central application.

The communications service is configured by creating a communications template file (CTF) using the NCR Communications Template Generator. The communications template file contains protocol dependent configuration parameters.

One of the parameters you define in the CTF is the name of the connection to use when sending messages to and receiving messages from Central. Although NDC+ did not know the name of the connection specified in the CTF, it was able to communicate with the Central application using an alias name of 'COMMUNICATIONS'.

Owing to the differences in the way Self-Service Support handles High Order communications compared with S4, the alias name 'COMMUNICATIONS' cannot be used in Advance NDC. Instead, the Application Core uses the connection name 'TPA CONNECTION'. This means when you create your CTF, you must define the name of the comms connection as 'TPA CONNECTION'. For details of creating a CTF for use by an SST running Advance NDC, refer to the *APTRA Communications Feature, User's Guide*.

Note 1: If the name you specify in the CTF for the connection is not 'TPA CONNECTION', a communications link will not be established.

Note 2: If you are using TCP/IP communications, you will not need to recreate your CTF as TCP/IP communications configuration is performed using the TCP/IP Configuration menus in Supervisor mode. For details, refer to the *APTRA Advance NDC, Supervisor's Guide*.

For information about registry and other settings that can be modified for communications, see "Configuring an Advance NDC Application" on page 5-2.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and/or distribution is strictly prohibited.

Migrating Existing NDC+ Exits

There are differences between Advance NDC and NDC+ in the way NDC+ Exits are handled. This section describes these differences and discusses some unsupported features and restrictions in Advance NDC you need to be aware of.

If you do not use NDC+ Exits, you can go straight to "Proving the Download Works" on page 3-25.

If your NDC+ Exits have added any ADI2 device-handling functionality they must be re-implemented to use the CEN-XFS interface. All devices are now supported through this interface.

For details of device access, see "Device Access" on page 4-7. For guidelines on migrating NDC Exits from ADI2 to CEN XFS, refer to the *CEN-XFS Exits in Advance NDC* white paper provided with Advance NDC. You can access the white paper from the APTRA on-line documentation. (Select **Start | Programs | NCR APTRA | APTRA (TM) Documentation** and in the APTRA Documentation Contents, under APTRA Advance NDC, select **Links to Publications**.)

Unsupported Features and Restrictions

This section discusses unsupported features and restrictions between Advance NDC and NDC+.

Hook Functions

Due to the differences in implementation, Advance NDC does not call the following Hooks:

- Clearing Fitness (Point-of-use 4 in MISCONT)
- Physically Clearing Devices (Point-of-use 5)
- Suspend Mechanism (Point-of-use 6)
- Process Communications (Point-of-use 7).

If these Hooks have been defined in the MISCONT Registry file, there will be no error; however the Hook will never be executed by Advance NDC.

Invoking an NDC+ Standard State from an Exit

The PerformNDCState function allows the execution of a standard NDC+ State Type from within your own Exit. This API (Application Programming Interface) is supported in Advance NDC.

NDC+ imposes the restriction that the Card Read State Types cannot be invoked directly from an Exit State.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

In Advance NDC, the following State Types cannot be invoked from within an Exit, as they have not been implemented as 'C' functions, but authored using the APTRA Author, or coded as C++ Classes:

- Card Read
- Card Read - PIN Entry Initiation
- PIN Entry
- Enhanced PIN Entry
- Transaction Request
- Close
- Customer Selectable PIN.

Invoking either of the Card Read State Types from within PerformNDCState will result in a return value of 4, as in NDC+.

Invoking any of the other unsupported State Types from within PerformNDCState will result in a return value of 1 (State Type Illegal).

The fact that these State Types cannot be called from within an Exit should not impact you greatly, given the increased flexibility Advance NDC offers.

Display Screens

NDC+ Supervisor C Exits that display screens cannot be easily ported to Advance NDC. These functions must either be authored using Advance NDC, or rewritten as follows:

- Using the TTU SP for the rear display
- Using the APIs for displaying NDC screens on the front display, as described in *NDC, Using NDC Exits.*

An example Exit is provided with Advance NDC; for details refer to the white paper, *CEN-XFS Exits in Advance NDC*, accessible from the on-line *APTRA Documentation* once Advance NDC has been installed.

Virtual Controller Restrictions

There are some restrictions on the use of virtual controllers in Advance NDC. These restrictions are explained below.

SendStatus and SendUnformattedData These callback functions can be called from an Exit, allowing you to send a message to Central. The ViaInterceptor parameter of these functions allows the messages to be optionally routed through Outgoing Virtual Controller Interceptors. This is supported in Advance NDC except for Stop and Delete.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

If you return a value of 1 (Stop and Delete) from the Outbound Interceptor for messages you have asked Advance NDC to send using the SendStatus and SendUnformattedData APIs, then Advance NDC will return a value of 1 (Failed Linked Up) to your call to SendStatus or Send UnformattedData.

Potential Transaction Reply Timeout Due To C Exit processing, if a virtual controller takes a long time to process a transaction reply message, Timer 3 may time out. The value of this timer should be increased accordingly to allow for the potential processing time.

DumpData Callback Function

Advance NDC has implemented this function, so there is no need to change your code, but no useful task is performed.

The usefulness of the function provided by NDC+ is limited. To check the value of any Shared Data, we recommend using a debugger.

Unsupported Shared Data

Any Exit code you have written to read or write to the Shared Data exposed by NDC+ will continue to work. The functions you call to access the data are the same in Advance NDC as in NDC+. Additionally, the identifiers you use to indicate which variable you wish to access are the same.

There is some data exposed by NDC+ that is not used. Although you can still read and write to these data items, Advance NDC does not use them. The unsupported data items either relate to features not supported in Advance NDC, or are specific to the way in which NDC+ has been implemented.

Using the File Management Interface

The *Using NDC Exits* publication promotes the use of the File Management Interface (FMIF). The FMIF provided as part of S4, is not provided in Self-Service Support. Therefore, any calls made to the FMIF for File Management will have to be changed.

Differences that do not Impact Exit Code

This section discusses the differences in implementation that do not impact Exit code.

Errors Loading an Exit

When NDC+ is unable to access an Exit, for whatever reason, it prints a message on the Journal. These errors are non-recoverable programming errors introduced by the Exit developer. They should be flagged at the earliest opportunity (equivalent to a Compile time

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

error), rather than waiting for a particular path through the application to be exercised (equivalent to a runtime error).

When Advance NDC cannot load a DLL, or find the function in the DLL to call, the MISCONT file is ignored and no errors are displayed or logged. The main indication of the problem is that the function defined in MISCONT is not called. The fix is to correct the definition.

Registry File Formats

In order to make these files more readable and easier to edit, new-line characters as well as 01D Hex are acceptable group separator characters. Also, one or more new-line characters may be substituted for the existing field separator character 01C Hex.

It is possible to implement this enhancement while maintaining compatibility with any existing Registry files which have been written. This is because a new-line character is not a valid entry in any of the fields preceding group or field separators in any of the Registry files used by NDC+.

STCONT Format

The format of the STCONT Registry file is slightly different in Advance NDC. There are no Group Separator characters between the following fields:

- State Type Character and the Support Flag
- Support Flag and the DLL Name.

Furthermore, the STCONT Registry file provided with Advance NDC has DLL and Function Names specified for most of the standard State Types, whereas NDC+ leaves them empty.

From Advance NDC 3.02, if the state parameter is a state number, the limit is set to ZZZ. Nested extension states are also supported, as described in “Primary and Extension State Tables” on page 3-15.

Editing STCONT The default STCONT registry file (STCONT) is located in the *<global>\test\xfs\dll* and *<global>\final\xfs\dll* directories on your development PC. It contains details of all valid State Types, and for each State Type indicates:

- Whether it is authored in Advance NDC or is a user-written Exit
- The number of the associated State Table entries and the maximum value of each of these entries
- Which State Table entries are references to extension States.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

The STCONT registry file consists of a number of entries, one for each State Type. The structure of each entry is as follows:

Table 3-4
STCONT Registry File Structure

Field Description	Size	Value
State Type Character	1 Byte	Example - 'M'
Advance NDC or Exit Support Flag	1 Byte	'N' or 'E' *See* "Advance NDC or Exit Support Flag" on page 3-15
DLL Name	VAR	*See* "DLL / Function Name" on page 3-16
Group Separator	1 Byte	1DH
Function Name	VAR	*See* "DLL / Function Name" on page 3-16
Group Separator	1 Byte	1DH
Primary State Table Information	VAR (Max. 8 sets of 4 Bytes)	*See* "Primary and Extension State Tables" on page 3-15
Group Separator	1 Byte	1DH
1st Extension State Table Max Limits	VAR (Max. 8 sets of 4 Bytes)	*See* "Primary and Extension State Tables" section
Group Separator	1 Byte	1DH
2nd Extension State Table Max Limits	VAR (Max. 8 sets of 4 Bytes)	*See* "Primary and Extension State Tables" on page 3-15
Group Separator	1 Byte	1DH
3rd Extension State Table Max Limits	VAR (Max. 8 sets of 4 Bytes)	*See* "Primary and Extension State Tables" on page 3-15
"	"	"
"	"	"
Group Separator	1 Byte	1DH
Last Extension State Table Max Limits	VAR (Max. 8 sets of 4 Bytes)	*See* "Primary and Extension State Tables" on page 3-15
Field Separator	1 Byte	1CH

You should add your Exit State Types to the STCONT file provided with Advance NDC. NCR recommends you take a copy of the original file before making changes as editing the NDC+ registry files is potentially error prone.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Group separator characters must be included up to the point where there is no more data to follow for any entry in question.

Note: If you modify the STCONT file, copy the modified file to both the *<global> \ test \ xfs \ dll* directory and the *<global> \ final \ xfs \ dll* directory. Additionally, copy the file to your custom directory, to be included in the aggregate when you export it from the Aggregate Builder Tool for installation on an SST

Advance NDC or Exit Support Flag If this field contains an 'E', this means the code for the State Type is supplied by an Exit. A value of 'N' means that the code for the State Type is provided by Advance NDC or an Advance NDC authored State Type. Other values are regarded as 'N'.

Although you may have re-implemented an NDC+ State Type using an Exit or an Advance NDC authored State Type, we recommend that you do not override the Card Read, Transaction Request and Close State Types in Advance NDC. To override any other State Type, you need to do the following:

1. Change the 'Advance NDC or Exit Support Flag' field to 'E' for an Exit, or 'N' for an Advance NDC authored State Type.

2. Change the name of the DLL / function name associated with the State Type.

Primary and Extension State Tables The Primary State Table, unlike the Extensions, has up to 8 sub-fields of 5 bytes.

The first 4 byte sub-field indicates the maximum limit for the associated entry. ASCII characters '0' to '9' should be used for this sub-field's bytes.

The second single byte sub-field indicates whether the entry is actually a reference to an Extension State associated with the Primary State. A value of '0' means that the entry is not a reference. A value of '1' to '8' indicates 1st, 2nd, 3rd...8th Extension reference. An entry with the value '1' to '8' will result in the matching limit being replaced by the maximum State number supported by the current release. Any value other than the nine described will result in the default Close State being invoked if any attempt is made to use the State Type definition during the State Flow.

From Advance NDC 3.02, nested extension tables are supported. This means that the Primary state defines one extension state and the Extension state defines another Extension state. You can define up to eight extension states.

The Extension state table has up to 8 sub-fields of 4 bytes. Extensions may be added up to the maximum possible for a base

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

State. They must be terminated by a Group Separator character or a Field Separator within 8 bytes. To support nested extension states, the first byte defines the next extension state. A value of '0' means no nested extension state is defined. The next three bytes define the maximum limit for the state or the next extension state number. The limit will be ZZZ if using alphanumeric state numbers. You can define a nested extension state on any sub-field in the extension state.

If a Group or Field Separator is detected before the 8th field, this will result in the remaining validation limits being set to zero. The Field Separator terminates the definition for the State Type. If it appears before the end of the 8 fields of Primary or Extension data, the remaining fields will be set to zero. Entries that are defaulted in this way will not be regarded as references to Extension States in the case of the Primary State Table.

DLL/Function Name The DLL Name can be specified using its full pathname; however, no DLL file extension should be supplied. Ensure the pathname and DLL file are specified using the correct case.

SUPCTR File Content

In NDC+, the SUPCTR file is released with entries for the full set of standard Supervisor functions.

In Advance NDC, the standard Supervisor functions are authored and not coded as 'C' functions. As a result, the SUPCTR file released with Advance NDC is empty.

As a consequence of this change, if errors are encountered when trying to read the Registry file, it is still possible to provide the default Supervisor behaviour, unlike in NDC+ Release 6, where 'Leave Supervisor' is the only option available.

You should add your Exit Supervisor menus/functions to the SUPCTR file provided with Advance NDC. If you have written C Exits that handle Supervisor screen display and keyboard input, these are not easily ported to Advance NDC, so this is not recommended. For further information, *see* "Specification of Exit Supervisor Menus" on page 3-18.

The SUPCTR file is located in the *<global>\test\xfs\dll* and *<global>\final\xfs\dll* directories on your development PC.

Note: If you modify the SUPCTR file, copy the modified file to both the *<global>\test\xfs\dll* directory and the *<global>\final\xfs\dll* directory. Additionally, copy the file to your custom directory, so that the modified file is included in the aggregate when you export it from the Aggregate Builder Tool for installation on an SST.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and/or distribution is strictly prohibited.

SUPCTR Error Handling

In NDC+, if there are errors in the definition of the SUPCTR file, an error message number is displayed to the operator at runtime, for example, E0042.

These errors are non-recoverable programming errors introduced by the Exit developer. They should be flagged at the earliest opportunity.

In Advance NDC, such an error results in the SUPCTR file being ignored and no errors are displayed or logged. The main indication of the problem is that the function defined in SUPCTR is not called. The fix is to correct the definition.

VCCONT File Content

The VCCONT file released with Advance NDC is empty, as there are no Inbound or Outbound Interceptors by default.

To register your Virtual Controller functions, add them to the VCCONT file provided with Advance NDC.

The VCCONT file is located in the *<global>\test\xfs\dll* and *<global>\final\xfs\dll* directories on your development PC.

Note: If you modify the VCCONT file, copy the modified file to both the *<global>\test\xfs\dll* directory and the *<global>\final\xfs\dll* directory. Additionally, copy the file to your custom directory, so that the modified file is included in the aggregate when you export it from the Aggregate Builder Tool for installation on an SST.

Differences that Impact Exit Code

This section discusses the differences in implementation that do impact Exit Code.

Definition of Variables

In a Windows environment, any global variables or variables defined with a ‘static’ keyword are allocated memory on a ‘per process’ basis. This could result in different Exit DLLs accessing different instances of the variable.

To avoid this, you must define a named memory section for these variables. For further information, refer to your Microsoft Visual Studio 2005 documentation.

Memory Allocation and De-allocation

In some cases for Windows, your Exit code may have needed to allocate memory which Advance NDC has to delete. Also, your Exit code may have to delete memory allocated in Advance NDC.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

The *Using NDC Exits* publication directs you to use ISO standard functions supplied with the Compiler for allocating and de-allocating memory. This is particularly relevant if you are using Virtual Controllers. However, the mechanism prescribed in the *Using NDC Exits* manual is not guaranteed to work, as different compilers implement memory management in different ways.

To combat this problem in Windows XP, we have provided functions you must call to Allocate and Free memory. These functions are provided in *exutil.lib* and *exutil.h*.

The Signatures of these functions are shown in the following table.

Table 3-5
Allocate and Free Memory Function Signatures

Signatures	Function Prototype	Usage
AllocateMessageBuffer	char * AllocateMessageBuffer (unsigned int nNumChars)	Used whenever the Virtual Controller needs to allocate memory for a message, ensuring consistent memory management throughout the application. AllocateMessageBuffer will allocate memory using the same method Advance NDC uses to create message buffers. If the allocation fails, NULL is returned. If the Virtual Controller returns a message buffer which was not allocated using AllocateMessageBuffer, an mError is invoked. It is essential that memory is allocated using this API in order that Advance NDC knows the size of the buffer returned from the Virtual Controller.
FreeMessageBuffer	void FreeMessageBuffer (char *pchBuffer)	Used whenever the Virtual Controller needs to free a message buffer (such as after copying a message to a larger buffer), ensuring consistent memory management throughout the application.
FreeStringVal	Void FreeStringVal ( )	Used to free memory allocated by GetStringVal

Specification of Exit Supervisor Menus

In NDC+, Menu Numbers 7, 8 or FFH can be used for Exit Supervisor Functions, as the SUPCTR file does not use the Function Number field for these menus.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

In Advance NDC, you must set the Function Number to FFH, for Menu numbers 7, 8 and FFH. If this is not adhered to, Exit Supervisor menus will not be invoked.

Exit Supervisor Menus

In addition to adding a new Exit Supervisor Function to an existing NDC menu, it is possible to add a new Exit menu which is accessible from an existing NDC menu. For Supervisor screen and keyboard handling, you must use an XFS interface that complies with CEN-XFS 3.

For more information about using the CEN-XFS interface in Advance NDC Exits, refer to the *CEN-XFS Exits in Advance NDC* white paper provided with the Advance NDC documentation.

Example As an example, we will assume you wish to add your own Exit Supervisor menu, to be entered by selecting '25' from the Select menu.

To do this, you would carry out these steps:

1. Edit the *resrvd.def* file by adding '`25 Exit Supervisor`' to reserved screen M00 (the Select Menu). Advance NDC will then handle the entry of '25' from the keyboard.

   The *resrvd.def* file is located in the <*global*> \ *test* \ *xfs* \ *dll* directory on the development PC. After making changes, copy the modified file to both the <*global*> \ *test* \ *xfs* \ *dll* and <*global*> \ *final* \ *xfs* \ *dll* directories. Additionally, copy the file to your custom directory to be included in the aggregate when you export it from the Aggregate Builder Tool for installation on an SST.

2. Write an Exit Supervisor Function to execute each of the functions within the menu, using the CEN-XFS interface (TTU and PIN service providers) to handle the screen and keyboard: display the menu screen, gather keyboard data, re-display the menu and so on. Alternatively, use the APTRA Author for the screen and keyboard handling as described in the next section, "Alternative Method" on page 3-20.

   Note: Any existing Exits that handle screen and keyboard data must also be changed to use the TTU and PIN service providers.

   The Menu Exit Supervisor Function should finish by setting up the next MenuId to display on exit from your own menu.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

3 Define the Supervisor Exit for the Exit Menu in the SUPCTR registry file as being Menu 0 (the Select Menu), Function Number 25. The Code in the Exit Menu Function would then set up the MenuId parameter to be 0 on return (assuming that you wanted to go back to the Select Menu).

When you specify your own Exit functions for invocation from your own Exit Supervisor Menu, these functions are called directly by the code you write in the Menu Function. These functions are **not** called as Supervisor Exits, so there is no need to define them in the SUPCTR file.

Of course, an Exit Supervisor Menu itself is just a special case of a Supervisor Function, one that needs to display a screen and get input from the keyboard.

For more detailed information, refer to the *APTRA Advance NDC, Reference Manual*.

Alternative Method

An alternative to developing your own Exit Supervisor Menu using NDC Exits, is to modify the Supervisor functionality in the Advance NDC Supervisor Application using the APTRA Author.

Using the APTRA Author to add a new menu from the Select Menu would be less work than writing 'C' code to display a screen and obtain keyboard input.

Importantly though, you can still re-use the 'C' code you have developed for your specific Supervisor Functions, and access them from a new authored menu. Within the APTRA Author, you can specify that an existing Supervisor Exit Executor worker should execute any pre-defined function for you.

Differences in the Development Process

This section discusses differences in the process used to develop Exits.

Registry File Locations

In NDC+, the Registry files, for example STCONT, are located in the <*system*> directory.

In Advance NDC, all Registry files are expected to be in the <*global*> \ *test* \ *xfs* \ *dll* and <*global*> \ *final* \ *xfs* \ *dll* directories. If you change any of these files, also copy the modified files to your custom directory, so that they are included in the aggregate when you export it from the Aggregate Builder Tool for installation on an SST.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Filename Differences

Any Exit DLLs that have been implemented and require migration to Advance NDC need to be recompiled in Microsoft Visual Studio 2005 service pack 1.

Although the same Callback functions are available in Advance NDC as in NDC+, the names of the files we release are different.

For example, where in OS/2 you included the following header files:

```
#include "..\incos2\ndcdata.h"
#include "..\incos2\s4adi2.h"
```

In Windows XP, these must be replaced by the following in *<global>\final\include*:

```
#include "ndcdatac.h"
#include "ndcdataf.h"
```

You must also include the CEN XFS header files relating to the device command sent from your exit.

The following table shows which library to link with and which header to include for each Callback.

Table 3-6
Callback Link Libraries and Headers

Callback function	Library to link with	Header to include
DisplayScreen	*ndccust.lib*	*ndccustm.h*
DumpData	*exutil.lib*	*exutil.h*
PerformNDCState	*exutil.lib*	*exutil.h*
RetrieveKeyboard	*ndccust.lib*	*ndccustm.h*
RetrieveScreen	*ndccust.lib*	*ndccustm.h*
SendStatus	*exutil.lib*	*exutil.h*
SendUnformattedData	*exutil.lib*	*exutil.h*
GetIntVal	*ndcdata.lib*	*ndcdataf.h*
GetStringVal	*ndcdata.lib*	*ndcdataf.h*
PutIntVal	*ndcdata.lib*	*ndcdataf.h*
PutStringVal	*ndcdata.lib*	*ndcdataf.h*
AllocateMessageBuffer	*exutil.lib*	*exutil.h*
FreeMessageBuffer	*exutil.lib*	*exutil.h*

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and/or distribution is strictly prohibited.

Callback function	Library to link with	Header to include
SetDigitalAudioPath	*exutil.lib*	*exutil.h*
PrintToJournal	*exutil.lib*	*exutil.h*
FreeStringVal	*ndcdata.lib*	*ndcdataf.h*

You should include the file *ndcdatae.h* if you make use of the Shared Data accessor functions. This file contains all of the identifiers for the data items.

Compiling Exit DLLs

Due to the way Exit functions are specified, they must be linked in the Exit DLL with 'C' linkage. The reason is that the user-supplied Registry files specify a function name without specifying parameters, therefore it is not possible to distinguish between overloaded versions of a function.

This can be done using the Microsoft Visual Studio 2005 compiler by specifying **extern "C"** and the Microsoft specific `_declspec(dllexport)` on each function definition, for example:

```
extern "C" void _declspec(dllexport) SupEntry(unsigned
char nInterface)
```

Naming Conventions

As with NDC+, the naming conventions of DLLs are not case sensitive.

In NDC+, the function names in Exit Code and in Registry files had to be uppercase. In Advance NDC, the function name in the Header, Source and Registry files must be the same case. That is, what was done for NDC+ is still compatible, but you are not restricted to use only uppercase.

Building DLLs

We do not impose any special requirements on you when building an Exit DLL, other than the information given in "Naming Conventions" on page 3-22.

You need to copy the built DLLs to your custom directory, to be included in the aggregate when you export it from the Aggregate Builder Tool for installation on an SST.

Useful Information

This section contains additional information to complement the *Using NDC Exits* publication.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Virtual Controllers - Use of Stop and Delete

It is important to understand the implications of returning a value of 1 from a Virtual Controller to NDC+.

Although you can return 'Stop and Delete' on a message other than an Unsolicited Status message, you should not do this, as it is likely to cause problems. It is your responsibility to use 'Stop and Delete' with care.

If, for example, 'Stop and Delete' is returned to a Transaction Request message, it would not be sent to Central and a timeout would occur.

Decomposing Messages in the Virtual Controller

The *Using NDC Exits* publication does not offer advice on how to parse messages sent from NDC+ to a Virtual Controller.

As Advance NDC and NDC+ apply the Security Fields at different times, NCR recommends that field separators be used to navigate through the message (as an NDC+ Central application would do), and not rely on specific lengths of data between any given field separator.

In NDC+, the message given in the Virtual Controller does not include fields such as the LUNO, Message Coordination Number and so on, although all field separators are present. In Advance NDC, the message given will include all of the Security Fields, with the exception of the MAC (Message Authentication Code).

Using StoreScreen

As described in the *Using NDC Exits* publication, the routine `StoreScreen` lets an Exit store a screen in the NDC+ screen pool.

In NDC+, `StoreScreen` has three parameters: Screen Group, Screen ID and Screen Data.

In Advance NDC, `StoreScreen` has two parameters: Screen Data and Screen Data Length.

The first parameter, Screen Data contains the following bytes:

Table 3-7
Screen Data Byte Structure

Byte Position	Format	Content
1	ASCII	Screen group
2 and 3	Binary (WORD)	Screen number
4....*n*	ASCII	Screen data

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

The second parameter, Screen Data Length, is the total length of the first parameter.

Multiple Exit Hooks

Multiple exit hooks can be used in Advance NDC.

To use one exit hook, no change is required. If you wish to use multiple exit hooks, see “Multiple Exit Hooks” in Appendix A, “Using Exits in Advance NDC”.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Proving the Download Works

After you have made the necessary changes to migrate graphics, audio and animation files and NDC+ Exits, you may want to install Advance NDC on an SST to confirm that it works correctly with your NDC+ download.

For details of how to install an Advance NDC application on an SST, see Chapter 10, “Delivering an Advance NDC Application to the SST”.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Chapter 4

Upgrading from Earlier Releases of Advance NDC

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and/or distribution is strictly prohibited.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Overview

The structure of the Advance NDC application has not changed since release 2.05. Wherever possible, existing functionality is unchanged.

This chapter describes the differences you need to be aware of when upgrading your Advance NDC installation from an earlier version, as follows:

- Feature support
- Device access and data sources
- Message handling and printing
- Configuration

For more information about the structure of Advance NDC, see Chapter 1, “Introducing Advance NDC”.

For information about configuration of Advance NDC see Chapter 5, “Configuring Advance NDC and Support Applications”

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Feature Support

This section describes changes in the implementation, functionality or operations of features compared to earlier versions of Advance NDC. For details of differences that apply only on other vendor's SSTs, refer to the *APTRA Advance NDC, Multi-Vendor Support Reference Manual.*

Changes to the Interface since Release 2.6

Devices that were previously accessed through ADI2 or CEN-XFS 2 are now accessed through the CEN-XFS 3 interface.

The content of messages is derived from the XFS data and mapped to the equivalent NDC message. In cases where an exact match is not possible, the closest compatible message is generated.

Access to the BNA device was through ActiveX, but now the ActiveXFS interface is used. Access to the CPM device was through ActiveX, but now the CEN-XFS interface is used. Access to the diskette drive and cardholder display is unchanged, through the operating system.

ADI2 to CEN-XFS 3

In Advance NDC 2.06, the following devices were accessed through ADI2:

- Swipe Reader
- 3 Track Write Reader
- Spray Dispenser
- Alarms
- Night Safe Depository
- Statement Printer
- Encryptor.

CEN-XFS 2 to 3

In Advance NDC 2.06, the following devices were accessed through CEN-XFS 2. Prior to Advance NDC 2.06, they were supported through ADI2:

- MCRW Tk123
- Smart MCRW
- Dip
- Smart / DIP Card Readers (without smart functionality)
- Cardholder and Operator Keyboards
- Cash Dispenser

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

- Envelope Dispenser and Depository
- Receipt Printer
- Journal Printer
- TI Bins (Alarms)
- System Display
- Media Entry Indicators.

For details of access to all devices in Advance NDC, see "Device Access" on page 4-7.

Communications

In Advance NDC, the default setup for communications is TCP/IP, which is configured through the Supervisor Access menu. For PCCM, or CCM VISA2 dialup communications, the registry must be edited. For details of all registry keys for communications, see "Configuring Communications" on page 5-11.

RSA Initial Key Loading

In Advance NDC 2.06, support for RSA was introduced. On SSTs with an EPP, the initial encryption keys (A key, B key and V key) can be downloaded from Central using RSA encryption instead of being entered locally in Supervisor mode.

Before using RSA encryption, the Host Security Module (HSM) on the host and the EPP on the SST must perform an authentication process that meets the requirements described in the "Security Features" chapter in the *APTRA Advance NDC, Reference Manual.*

Additional BNA Functionality

In Advance NDC 3.0, as well as including transaction counts in the transaction request message, option 45 of the Enhanced Configuration Parameters Load is used to configure the following:

- Reporting of retract operations
- Number of notes to accept (up to 90 or more than 90)

Advance NDC 3.01 also includes the reporting of individual cassette status through an extended message format.

Enhanced Night Safe

In Advance NDC 2.06, the enhanced night safe was supported.

From version 3.0, the enhanced night safe is not supported, as the variant information is unavailable.

Enhanced Configuration Parameters Load

Enhanced Configuration Parameters Load messages are handled as previously, except for MCRW Enhanced Card Drive (ECD) Security Jitter.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and/or distribution is strictly prohibited.

Option number 46, which is used to set the registry key for MCRW Enhanced Card Drive (ECD) Security Jitter, takes effect only when the SST is reset.

Note: If this option is set, any card reader security options set outside Advance NDC, such as through the Card Reader Property pages, are lost.

BAPE Emulation

The EPP in BAPE emulation mode is not supported from release 3.00 of Advance NDC; therefore the full EPP functionality must now be used.

EMV C Exits

As in Advance NDC 2.06, the Advance NDC application will operate with existing EMV C Exits that are EMV Level 2 compliant. EMV/CAM2 Exits for APTRA Advance NDC release 2.00.01 must be used.

Keyboard Data

From Advance NDC 2.06 onwards, the following configuration parameters are not supported:

- Downloadable keyboard definitions. If different keyboard layouts are required, they must be specified through the PIN service provider.
- Cancel/Clear swap option.

Note: The FDK swap option is supported as it applies only to the front keyboard and to some state types that require dynamic swapping at application runtime.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and/or distribution is strictly prohibited.

Supervisor Menus

From Advance NDC 3.00, Supervisor has option 38 on the Configure menu leading to submenus for dialup communications functions. The options on the Dialup Config menu are as follows:

Table 4-1
Dialup Config Menu Options

Menu	Description
1 APPL PARAM	Submenu to configure application parameters for sending messages
2 MODEM LINK	Submenu to configure modem parameters: telephone numbers, dial mode and modem timers
3 NETWORK ADD	Submenu to configure network address and timers
4 SERIAL LINK	Submenu to configure serial properties
5 VISA 2	Submenu to configure CCM VISA2 connection timers
6 COPY ON / OFF	Menu to transfer dialup application and modem configuration to or from diskette
7 DIAGNOSTICS	Menu to view and test modem configuration

From Advance NDC 3.01, alphanumeric entry can be used for certain functions.

For full details of the dialup menu functions, refer to the *APTRA Advance NDC, Supervisor's Guide*. For more information about configuring dialup communications, see "Configuring Communications" on page 5-11.

Fault Display

While an SST is in service, messages about device faults are displayed cyclically on the rear operator panel if one is attached. These messages give information about devices requiring attention soon, such as Receipt Low, and devices in a fatal condition, such as Cash Handler Fatal. The messages also give details of the corrective measures necessary to maintain an operational SST.

When the SST is in Supervisor mode, the fault display can be displayed on request using one of the supervisor functions. This applies to both front and rear interfaces.

Note: Fault display is not supported for BNA or DASH devices.

In Advance NDC 2.06, the means of selecting the fault display supervisor function changed. Previously, the FDK H key was used to invoke fault display. From Advance NDC 2.06, the SELECT menu option 26 - FAULT DISPLAY is used. The CANCEL key is used to exit the Fault Display and return to the SELECT menu.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

As all the device status information is provided through XFS, fault display messages indicate the device or device type and, where possible, give extra details such as "`CASH HANDLER FATAL, SHUTTER JAMMED`" to enable the operator to identify the failure.

The Fault Display shows all faults on one screen if possible, using additional screens if necessary. The operator is given enough time to read them as the value of Timer 92 is extended based on the number of text lines displayed at a time. If there are no faults, the fault display shows reserved screen `E1903 NUMBER OF FAULTS = 0`.

The text strings for fault messages can be localised by editing *resrvd.def*. They are contained in the screen group *Ennnn*.

User Messages and User Terminal Data Implementation

Before Advance NDC 2.06, User Messages and user Terminal Data were implemented as ActiveX controls. In Advance NDC 2.06, they were re-implemented in the Application Core, with unchanged functionality. For information about these changes and their impact on customised applications, see "User Messages and User Terminal Data" on page 4-12.

APTRA Simulator

For information about how to obtain the APTRA Simulator, refer to the Advance NDC release bulletin. For information about using the simulator, see "Testing the Advance NDC Application" on page 6-32, and refer to the release bulletin.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Upgrading from Earlier Releases of Advance NDC

If you use the Advance NDC application as provided by NCR, you will be able to upgrade with only a few configuration changes, described in "Configuring an Advance NDC Application" on page 5-2.

As devices are now accessed through the XFS interface instead of ADI2, changes will be required if you have done any or all of the following:

- Added any new ADI2 device-handling functionality
- Applied customisations to Author applications that include device worker support and the workers are no longer supported or their behaviour has changed
- Created your own User Messages or added your own data to Terminal State messages.

The device status information is maintained for compatibility with previous versions of Advance NDC.

Worker Class Support

If your application uses workers that either function differently or are not supported in multi-vendor Advance NDC, you will need to assess the extent of the changes required before you migrate your application. An alternative to using an unsupported worker is to use an authored exit state.

For more information about worker class support in Advance NDC, building an application and the locations for customised files, refer to the *APTRA Author, User's Guide.*

C Exits

C Exits are supported in a multi-vendor environment, though C Exits written for an earlier release of Advance NDC will need modification if they access devices.

EMV/CAM2 Exits for APTRA Advance NDC

For EMV Exits support, you must use release 2.00.01 of the product for CEN-XFS 3.

Device Access

The following table gives details of the service providers used for device access. Device identifiers are present in terminal configuration Terminal State messages. They indicate the device to which the data in the message applies. For some supported variants,

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

it is not always possible to identify each unique variant, for example, the DIP readers.

For details of the hardware configuration data associated with each device, refer to the *APTRA Advance NDC, Reference Manual,* Appendix H, “Device Identifiers”.

Table 4-2
Service Provider Device Access Details

Device ID	Device Name	Device Variant	Device Access	Comment
D	Magnetic Card Reader/Writer	Track 2	IDC SP	Read only
		Track 1/2/3 MCRW	IDC SP	Write on track 3 only
		PC Dip Reader	IDC SP	—
		3 Track Write MCRW	IDC SP	—
		Track 2 Smart Card Reader	IDC SP	—
		Track 1/2/3 Smart Card Reader	IDC SP	—
		3 Track Write Smart Card Reader	IDC SP	—
		Track 2/3 Dip MSR	IDC SP	Variant returned as Track 2/3 Reader
		Track 1/2 Dip MSR	IDC SP	—
		Track 1/2 Swipe Reader	IDC SP	—
		Dip and Smart Hardware () Card Reader	IDC SP	—.
E	Cash Handler	Standard cash handler	CDM SP	—
		Spray dispenser	CDM SP	—
		Dual cash handler	CDM SP	—
F	Envelope Depository		DEP SP	—
G	Receipt Printer	Dot matrix	PTR SP	—
		Thermal	PTR SP	—
H	Journal Printer	Dot matrix	PTR SP	—
		Thermal	PTR SP	—
K	Night Safe Depository		DEP SP	Only Basic Night Safe variant supported
L	Encryptor	NBS encryptor	PIN SP	This is reported for BAPE/HI-BAPE

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and/or distribution is strictly prohibited.

Device ID	Device Name	Device Variant	Device Access	Comment
		EPP encryptor	PIN SP	—
O	Off-Line Flex Disk	1.44 Mb Flex Disk Drive	O/S	Access to the diskette drive is through the operating system and is vendor-independent
		2.88 Mb Flex Disk Drive	O/S	
P	TI Bins (Alarms)	All variants	SIU SP	—
Q	Cardholder Keyboard	Standard (BAPE) keyboard	PIN SP	—
		EPP keyboard	PIN SP	—
R	Operator Keyboard	Keyboard plus FDKs (enhanced)	TTU SP	This is reported for the Enhanced Operator Panel
S	Cardholder Display		O/S	Access to display, touch screen, and voice functions is through the operating system and is vendor-independent
V	Statement Printer	All variants	PTR SP	—
Y	Coin Dispenser	All variants	CDM SP	—
Z	System Display	Enhanced display (16 x 32)	TTU SP	—
[	Media Entry Indicators		SIU SP	—
\	Envelope Dispenser		DEP SP	—
q	Cheque Processing Module (CPM)	All variants	PTR SP	Supported only on NCR SSTs
w	Bunch Note Acceptor (BNA)	All variants	ActiveXFS	Supported only on NCR SSTs

Supplies Data Sources

The following table gives details of the information source for device supplies data.

For details of the supplies data associated with each device, refer to the *APTRA Advance NDC, Reference Manual,* Appendix H, “Device Identifiers”.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and/or distribution is strictly prohibited.

Table 4-3
Supplies Data Source Information

Device ID	Device Name	Supplies Data	Data source
D	Magnetic Card Reader / Writer	Card capture bin	XFS IDC
E	Cash Handler	Cash handler reject bin	XFS CDM
		Cash handler cassette type 1	XFS CDM
		Cash handler cassette type 2	XFS CDM
		Cash handler cassette type 3	XFS CDM
		Cash handler cassette type 4	XFS CDM
F	Envelope Depository	Envelope deposit bin	XFS DEP
G	Receipt Printer	Receipt printer paper	XFS PTR
		Receipt printer ribbon	XFS PTR
		Receipt printer print-head	DSM
		Receipt printer knife	DSM
H	Journal Printer	Journal printer paper	XFS PTR
		Journal printer ribbon	XFS PTR
		Journal printer print-head	DSM
K	Night Safe Depository	Night safe bin	XFS DEP
V	Statement Printer *all variants*	All supplies data	XFS PTR
Y	Coin Dispenser	Coin dispenser module and all hopper types	XFS CDM
\	Envelope Dispenser	Envelope dispenser hopper	XFS DEP
w	Bunch Note Acceptor (BNA) *all variants*	All BNA cassettes	ActiveXFS

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Fitness Data Sources

The following table gives details of the information source for fitness data.

For details of the fitness data associated with each device, refer to the *APTRA Advance NDC, Reference Manual,* Appendix H, “Device Identifiers”.

Table 4-4
Fitness Data Source Information

Device ID	Device Name	Fitness Data	Data source
D	Magnetic Card Reader / Writer	Magnetic Card	XFS IDC
E	Cash Handler	Cash handler reject bin	XFS CDM
		Cash handler cassette type 1	XFS CDM
		Cash handler cassette type 2	XFS CDM
		Cash handler cassette type 3	XFS CDM
		Cash handler cassette type 4	XFS CDM
F	Envelope Depository	Depository	XFS DEP
G	Receipt Printer	Receipt Printer	XFS PTR
H	Journal Printer	Journal Printer	XFS PTR
K	Night Safe Depository	Night Safe	XFS DEP
L	Encryptor	Encryptor	XFS PIN
Q	Cardholder Keyboard	Cardholder Keyboard	XFS PIN
V	Statement Printer	Statement Printer	XFS PTR
Y	Coin Dispenser	Coin dispenser module and all hopper types	XFS CDM
\	Envelope Dispenser	Envelope Dispenser	XFS DEP
q	Cheque Processing Module (CPM)	CPM	XFS PTR
w	Bunch Note Acceptor (BNA)	BNA	ActiveXFS

Message Handling

For devices accessed through XFS, the message format is unchanged but the content of both solicited and unsolicited status messages is derived differently.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

The following table summarises the changes that apply to the devices listed earlier in this chapter under "Changes to the Interface since Release 2.6" on page 4-2. For further details of messages, refer to Chapter 9, "Terminal to Central Messages in the *APTRA Advance NDC, Reference Manual.*

Table 4-5
Message Handling Changes Summary

Status Descriptor	Field	Description	Comment
'**C' - Specific Command Reject**	g1 = 'C' g2 = 02	The message is not accepted while diagnostics, or clear SOH level 2 or level 3 is in progress.	This is returned when the application has passed control to Vendor Dependent Mode (VDM).
'F' - Terminal State			
1 - Send Configuration Information	g3 g5 g6	Device Severity Supplies Status Sensor Status	The value is derived from information provided by the SP.
'H' - Hardware Configuration Data	g4	Hardware Configuration	The device configuration is derived from the XFS capabilities.
'I' - Supplies Data	g2	Supplies Status	Supplies data is derived from XFS.
'J'- Fitness Data	g2	Fitness	Device fitness is derived from XFS.
'K' - Tamper and Sensor Status data	g2 g3	Sensor Status Tamper Status	The value is derived from XFS. As the DPM is not supported, character 13 is always set to 0.
'L' - Software ID and Release Number data	g2	Release Number Identifier	The value is 030000 for Advance NDC 3.00.
'8' - Device Fault	g3 g4 g5	Error Severity Diagnostic Status Supplies Status	The value is derived from XFS information.

For full details of the status message format and fields, refer to the *APTRA Advance NDC, Reference Manual.*

User Messages and User Terminal Data

If you have created new Message Classes or included additional data in Terminal State messages, you will have to import your code into the Application Core, where the functionality that used to be provided in separate ActiveX controls is now supported directly by the Message Handler for each mode.

You import your User Messages director or User Terminal Data director for each mode as required. The following table gives the

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

component ID of the Message Handler director where you insert your code.

Table 4-6
Message Handler Component ID

Mode	Message Handler Component ID
Out-of-Service	11L286B7
Supervisor	15L302B7
In-Service	69L302B7

Complete the following steps:

1. Make a copy of the Advance NDC application and open the Application Core project.
2. Import your User Messages director or User Terminal Data director from the old catalog into the Application Core.
3. Insert your User Messages or Terminal Data director under the Message Handler director for the required mode, replacing the default implementation of the User Messages director or the User Terminal Data director.

Printing

In Advance NDC 2.06, access to the receipt, journal and envelope printers was changed to the CEN-XFS interface. From Advance NDC 3.00, the statement printer is also accessed through the CEN-XFS 3 interface.

All print data is passed through a conversion DLL before an XFS call is made to the printer service provider using `WFS_CMD_PTR_RAW_DATA`. The NDC printer control sequences are still supported under the NDC message format and the existing workers.

Printer Variants As NCR service providers report text-only support, the default printer configuration is thermal. Where a non-thermal printer is configured, the registry can be set so that all unsupported commands are stripped before the print is sent to the printer. Graphics and barcode sequences are most likely to be unsupported by dot matrix printers.

Receipt Printer

To facilitate the printing of receipts, a new worker class was introduced in Advance NDC 2.06.

NDC Print Footer Worker This new worker prints to a receipt any remaining preprint information not printed by Print Buffer, and ejects the receipt. For details, refer to the Advance NDC Worker Class help in the Author.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Journal Printer

When hardcopy backup is set from the host (enhanced configuration option 16 or 17), status messages for the physical printer and hardcopy backup are reported to the host. When the EJ is selected by the user at the SST (`SET JRNL` option 1), messages are reported for the EJ.

If either the EJ or hardcopy backup log (*ejdata.log* or *hbdata.log*) reaches its size limit, and automatic INIT by cutover is not enabled, a fatal error message is sent and the printer enters a fatal state. A warning message is also sent when the EJ log is 90% full. For information on automatic INIT by cutover, see "Cutover" on page 5-37.

From release 3.01, the following EJ options are available:

- Enhanced EJ backup, providing the following choices:
  - — A single backup file using Standard EJ backup
  - — Up to 1000 backup files depending on disk space using Multiple EJ backup.

  For details, see "Enhanced EJ Backup" on page 5-36.
- Multiple destinations, allowing the choice of available destinations for the backup file. For details, see "Multiple Destinations for EJ Backup" on page 5-36.

From release 3.02, the following EJ options are available:

- The host control of the EJ backup mode can be disabled. For details, see "Disabling Host Control of Enhanced EJ Backup" on page 5-41.
- The INIT operation can be run automatically in a number of ways. An optional copy of the EJ file can also be made following the automatic INIT. For details, see "Automatic INIT Options" on page 5-37.
- The EJ file can be compressed. For details, see "EJ Compression" on page 5-40.
- A maximum size for the EJ file can be set. For details, see "Maximum EJ File Size" on page 5-41.

For information on using all EJ options, refer to the *APTRA Advance NDC, Supervisor's Guide*.

Dual Mode Operation From Advance NDC 2.06 onwards, dual mode operation is supported. When dual mode is selected by the user (`SET JRNL` option 2), only messages for the physical printer are sent to the host. No status messages for the EJ or hardcopy backup are sent unless enhanced configuration option 35 is set as follows:

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Table 4-7
Dual Mode Error Message Selection

Value	Effect in Dual Mode
000	Send unsolicited error messages for the physical journal printer only (the default value)
001	Send unsolicited error messages for the physical journal printer and EJ only
002	Send unsolicited error messages for both EJ and hardcopy backup

Note: When the journal printer is set to dual mode, hardcopy backup is automatically enabled. If option 16 or 17 is sent from the host, it will be ignored.

STCONT File Updates

From Advance NDC 3.02, the STCONT file supports alphanumeric state entries for up to 46655 state numbers.

The STCONT file has been modified to so that all state parameters, which refer to state numbers, have their limit set to ZZZ (base 36). For any new states, the limits of state parameters that represent state numbers must be set to ZZZ.

Nested extension states are also supported, as described in "Primary and Extension State Tables" on page 3-15.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Chapter 5

Configuring Advance NDC and Support Applications

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and/or distribution is strictly prohibited.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and/or distribution is strictly prohibited.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Overview

This chapter describes how to configure Advance NDC and any additional applications under the following headings:

- “Configuring an Advance NDC Application” (for runtime)
- “Silent Debug” (live environment)
- “DebugLog” (test environment)
- “The Custdat.exe Utility” (for creating test environments).

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Configuring an Advance NDC Application

This section describes the configuration settings that have been introduced since Advance NDC 2.06.

Configuration Options

From Advance 2.06, the following configuration options are not supported:

- Cancel / Clear swap option
- Download of keyboard definitions.

Advance NDC Registry Key

The registry root key for Advance NDC is as follows:

```
HKLM\SOFTWARE\NCR\Advance NDC
```

All registry settings for the product are in subkeys of this key.

Cash Handler Configuration

A currency cassette mapping table held in the registry of the SST is used to map the XFS logical cassette units to the Advance NDC cassette types. On NCR SSTs, the mappings between the physical cassette units, the XFS logical cassette units and the Advance NDC cassette types are automatically set up. Therefore, no additional operator configuration is needed after the initial installation of Advance NDC, nor any reconfiguration after replenishment.

Note: If either the CDM SP or the currency cassette mapping table is changed from the default, errors will occur or incorrect notes will be dispensed. This will be obvious from testing, or the errors generated. The default values are stored in *AdvanceNDC.reg* and *NCR_SPs.reg* in the DLL directory.

Setting Currency

The currency used by each cassette type is set using the following key:

```
HKLM\SOFTWARE\NCR\Advance NDC\CurrencyTable\NDCTypex\CurrencyID
```

Each cassette type must be defined using the *x* in NDCType*x* to identify the cassette type.

The value is the Currency ID in ISO format, for example EUR for Euro or GBP for sterling.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Setting Media Type

The kind of cassette in use is set using the following key:

```
HKLM\SOFTWARE\NCR\Advance NDC\CurrencyTable\NDCTypex\MediaType
```

Each cassette type must be defined using the *x* in NDCType*x* to identify the cassette type.

The content of this key is the CEN-XFS media type. For a GBRU, this must be set to 3. For details of the media types, refer to the APTRA on-line documentation for usType, under **APTRA XFS | Programmers Reference | XFS Service Providers**.

Setting Denomination

The value of the notes used by each cassette is set using the following key:

```
HKLM\SOFTWARE\NCR\Advance NDC\CurrencyTable\NDCTypex\Value
```

Each cassette type must be defined using the *x* in NDCType*x* to identify the cassette type.

Setting Multiple Currencies

Multiple currencies are defined by setting up all present cassette types in the currency cassette mapping table as described in “Setting Currency”, “Setting Media Type”, and “Setting Denomination” above. Cash handlers can then provide alternate currencies using the defined cassette types, as illustrated in the following examples:

- Set cassette types 1 and 2 up for Euro (EUR) denominations while cassette types 3 and 4 are set up for sterling (GBP) denominations.
- On dual cash handlers, if Option 76 supports up to four cassette types, cassette types 1 and 2 in the primary cash handler can be set up for EUR denominations while cassette types 3 and 4 are set up in the secondary cash handler for GBP denominations.
- On dual cash handlers, if Option 76 supports up to seven cassette types, cassette types 1, 2, 3 and 4 can be set up for EUR denominations while cassette types 5, 6 and 7 can be set up for GBP denominations. The cassette types can be in either cash handler as required.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Enabling Support for Seven Cassette Types

To configure support for seven cassette types, you must do the following:

1 Ensure that the Advance NDC currency cassette mapping table is correctly set up for seven cassette types, as described in "Setting Currency", "Setting Media Type", and "Setting Denomination" above.

2 Enable extended cassettes for the service provider by setting the following registry key to 1:

```
HKLM\SOFTWARE\Classes\WOSA/XFS_ROOT\SERVICE_PROVIDERS\CDM\
GENERAL_CONFIGS\EnableExtendedCassettes
```

Note: If using dual cash handlers, you must also set the following registry key:

```
HKLM\SOFTWARE\Classes\WOSA/XFS_ROOT\SERVICE_PROVIDERS\CDM2\
GENERAL_CONFIGS\EnableExtendedCassettes
```

3 Define a currency and value for cassette types 5 to 7. For example, to enable support for 10 GBP in cassette type 5 add the following keys:

```
HKLM\SOFTWARE\Classes\WOSA/XFS_ROOT\SERVICE_PROVIDERS\CDM\
GENERAL_CONFIGS\CurrencyIDType5=GBP

HKLM\SOFTWARE\Classes\WOSA/XFS_ROOT\SERVICE_PROVIDERS\CDM\
GENERAL_CONFIGS\ValuesType5=10
```

Note: This can be done using the *NCR_SPs.reg* file, which includes keys for cassette types five to seven commented out by default.

4 Set Enhanced Configuration option 76 to 001 to use extended messages for seven cassette types. For details of option 76, refer to the *APTRA Advance NDC, Reference Manual*.

5 At the SST, use the System Application to set up the note configuration for cassette types 5 to 7. For details of the System Application, refer to the *Self-Service Support, System Application User Guide*. You can access this PDF from the on-line APTRA Documentation: select **APTRA XFS | Runtime Maintenance | System Application**.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Setting Note Thresholds

You can set low and maximum thresholds for the number of notes in a cash handler.

Note: The maximum number of notes that can be dispensed is set by the hardware capabilities. For bunch dispensers, the MaxBills registry key can be set to either 40 or 50 in the configurable parameters for the CDM SP. For details, refer to the APTRA on-line documentation, under **APTRA XFS | Programmers Reference | XFS Service Providers**.

Low Threshold The threshold for notes taken from a cassette before reporting as low is set using the following key:

```
HKLM\SOFTWARE\NCR\Advance NDC\CurrencyTable\NDCTypex\
LowBillsThreshold
```

Each cassette type must be defined using the *x* in NDCType*x* to identify the cassette type.

The default is '0' and uses the hardware sensors. For details, refer to the APTRA on-line documentation for ulMinimum, under **APTRA XFS | Programmers Reference | XFS Service Providers**.

Maximum Threshold The threshold for notes deposited into the reject bin to trigger a device status of high is set using the following key:

```
HKLM\SOFTWARE\NCR\Advance NDC\CurrencyTable\RejectBin\
MaxBillsThreshold
```

The default is '0', which switches off the counter thresholds. For details, refer to the APTRA on-line documentation for ulMaximum, under **APTRA XFS | Programmers Reference | XFS Service Providers**.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Supplies and Severity Information

Supplies and severity information is updated to reflect the known state of the cash handler as described in Table 5-1.

Table 5-1
Supplies and Severity

Point at which update occurs...	Cassette Status	Severity Reported As...	Supplies Reported As...
Start of Day	Good	Good	Sufficient Notes
	Low	Good	Notes Low
	Empty	Good	Notes Low
	Missing	Good	No new state
Supervisor Exit	Good	Good	Sufficient Notes
	Low	Good	Notes Low
	Empty	Good	Notes Low
	Missing	Good	No new state
Following a TEST CASH operation	Good	Good	Sufficient Notes
	Low	Good	Notes Low
	Empty	Fatal	Out of Notes
	Missing	Good	No new state
Following a note dispensing transaction Status information is only updated for cassettes used in the transaction	Good	Good	Sufficient Notes
	Low	Good	Notes Low
	Empty	Fatal	Out of Notes
	Missing	Fatal	Out of Notes
All points above Status information reported for the Reject bin	Good	Good	No overfill
	Overfilled	Fatal	Overfill
	Missing	Good	No overfill

Dual Cash Handler Configuration

From Advance NDC 3.01, a second cash handler (known as the secondary cash handler) can be added to create one logical cash handler, which emulates a single cash handler for the host.

Note: In messages the cash handlers are identified using 0 and 1; in Advance NDC screens they are identified using 1 and 2.

The registry key for dual cash handler settings is:

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

```
HKLM\SOFTWARE\NCR\Advance NDC\DualCashHandler
```

The secondary cash handler has the following alias:

```
HKLM\SOFTWARE\NCR\Advance NDC\Aliases\Cash Dispenser Second
```

Setting Cash Handler Priority

The cash handler that is given priority is checked first and used to dispense whenever possible.

Note: If there is no primary cash handler attached to the SST, the secondary cash handler will never be available, even if it is set as the priority cash handler.

The following registry key contains the setting for cash handler priority:

```
HKLM\SOFTWARE\NCR\Advance NDC\DualCashHandler\Priority
```

Valid values for this key are as follows:

- 0 to give priority to the primary cash handler. This is the default
- 1 to give priority to the secondary cash handler.

Setting Counter Entry Mode

The counters for a dual cash handler can be set as combined or separated. If combined counters are used, enter the total count for both cash handlers using the ADD CASH Supervisor option. If separated counters are used, enter the count for each cash handler individually in ADD CASH. When the counts are displayed or printed, the total count for both cash handlers is used.

The following registry key contains the setting for combined or separated counts:

```
HKLM\SOFTWARE\NCR\Advance NDC\DualCashHandler\CountersEntryMode
```

Valid values for this key are as follows:

- 0 to use combined counters. This is the default.
- 1 to use separate counters.

Status Reporting

As each cash handler has a separate dispenser, each transaction uses one cash handler.

The status of dual cash handlers is reported as a single logical cash handler as follows:

- The severity is reported for the cash handler with the best status.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

- The fitness is reported for the cash handler with the best status or a suspend status. A suspend status is always reported whatever the fitness of the other cash handler. For further information, see "Cash Handler Status Message" on page 5-8.

Cash Handler Status Message

The following registry key defines whether to report a combined (best overall) or suspend status when the specified note mix cannot be dispensed by one cash handler:

```
HKLM\SOFTWARE\NCR\Advance NDC\DualCashHandler\CassSuspendedValue
```

Valid values for this key are as follows:

- 0 to report a combined status. This is the default.
- 3 to report a suspend status.

The following scenario illustrates the effect of this registry setting:

If only one of the cash handlers can dispense the requested note mix, the following occurs:

— The cash handler that can dispense the note mix is used.

— A solicited status message is sent if a cassette type runs out of notes during the dispense operation.

In this situation, neither cash handler can now provide the note mix required, and the status message reports a combined status by default. This reports the best overall status, which states that sufficient notes are available for the dispense. The host therefore continues to request a note mix that cannot be dispensed. If configured to report a suspend status, the host can continue requesting the cassette type as it is still available in one of the cash handlers, but will not request the note mix.

If both cash handlers could have dispensed the note mix, the priority cash handler is used. If the priority cash handler runs out of notes during the dispense operation, using the suspend setting means that the transaction can be re-attempted.

The following example illustrates this situation:

- Each cash handler contains cassette types 1 and 2
- Cassette type 1 is already fatal in the priority cash handler
- Cassette type 2 runs out of notes during a dispense using the second cash handler.

This situation means that a mix of cassette types 1 and 2 cannot be dispensed from either cash handler. However, reporting a combined status means that the supplies appear to be good because cassette type 2 is available in the priority cash handler.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Reporting a suspend status allows the host to continue requesting cassette type 2 as it is still available in the priority cash handler, but not in a mix with cassette type 1.

Interlock Handling

Dual cash handlers may each have a safe door interlock or there can be a single interlock that disables power to both cash handlers. If there are two interlocks on dual cash handlers that emulate a single cash handler, the interlock open M-Status can be reported with a non-fatal severity. To avoid this, interlock handling is defined by Enhanced Configuration option 76 as follows:

- If option 76 is set to 000, an open interlock is always reported as fatal
- If option 76 is set to 001, the best overall status is reported. The individual status is reported using DIGs ‘d’ and ‘e’ in response to a Send Configuration command. For information on this command, refer to the *APTRA Advance NDC, Reference Manual.*

Tamper Indication

As both cash handlers are reported to the host as a single logical cash handler, some messages to the host can seem to be duplicated. This may imply that tampering has occurred. If this is the case check the journal trace, which includes information on the cash handler.

For example, if all cassettes are removed from CH0 and then all cassettes are removed from CH1, the host message will suggest that cassettes have been removed twice. However, the journal trace prefixes this information with C*x*., where *x* indicates the individual cash handler.

For further information on tamper indication, refer to the *APTRA Advance NDC, Reference Manual.*

Spray Dispenser Configuration

If a spray dispenser is configured, NCR recommends that the MaxBills registry key be set to 70 in the configurable parameters for the CDM SP. For details, refer to the APTRA on-line documentation, under **APTRA XFS | Programmers Reference | XFS Service Providers**.

Coin Dispenser Configuration

A coin mapping table held in the registry of the SST is used to map the cash units to the Advance NDC coin hopper types. On NCR SSTs, the mappings between the physical hoppers, the cash units and the Advance NDC hopper types are automatically set up. Therefore, no additional operator configuration is needed after the initial installation of Advance NDC, nor any reconfiguration after replenishment. Advance NDC applies a one-to-one mapping of hopper type to cash unit.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Note: If either the CDM SP or the coin mapping table is changed from the default, errors will occur or incorrect coins will be dispensed. This will be obvious from testing, or the errors generated.

Setting the Currency

The currency dispensed by a hopper is set using the following key:

```
HKLM\SOFTWARE\NCR\Advance NDC\CoinTable\NDCHopperTypex\CurrencyID
```

Each hopper type must be defined using the *x* in NDCHopperType*x* to identify the hopper type.

Advance NDC provides eight entries, set by default to USD.

Note: The `CurrencyID` must be a valid ISO currency code.

Setting the Coin Value

The value of the coins dispensed by a hopper is set using the following key:

```
HKLM\SOFTWARE\NCR\Advance NDC\CoinTable\NDCHopperTypex\Value
```

Each hopper type must be defined using the *x* in NDCHopperType*x* to identify the hopper type.

Advance NDC provides eight entries, set by default as follows:

Table 5-2
Default Hopper Values

Type	Default Coin Value	Default Value in the Default Currency
1	10	10 cents
2	20	20 cents
3	50	50 cents
4	100	1 dollar
5	10	10 cents
6	20	20 cents
7	50	50 cents
8	100	1 dollar

Setting the Low Threshold

You can set low thresholds for the number of coins in a hopper. This defines the threshold for coins taken from a hopper before reporting as low.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Caution: Allowing the coin dispenser to become empty can cause jams.

This is set using the following key:

```
HKLM\SOFTWARE\NCR\Advance NDC\CoinTable\NDCHopperTypex\
LowCoinsThreshold
```

Where *x* identifies the hopper type. Include entries for each hopper type in use.

Advance NDC provides eight entries, set by default to 5.

Supplies and Severity Information

Supplies and severity information is updated to reflect the known state of the coin dispenser as described in Table 5-3.

Table 5-3
Supplies and Severity

Point at which update occurs...	Cassette Status	Severity Reported As...	Supplies Reported As...
Start of Day	Good	Good	Sufficient Coins
	Low	Good	Coins Low
	Empty	Good	Coins Low
	Missing	Good	No new state
Supervisor Exit	Good	Good	Sufficient Coins
	Low	Good	Coins Low
	Empty	Good	Coins Low
	Missing	Good	No new state
Following a CHECK COIN operation	Good	Good	Sufficient Coins
	Low	Good	Coins Low
	Empty	Fatal	Out of Coins
	Missing	Good	No new state
Following a coin dispensing transaction Status information is only updated for hoppers used in the transaction	Good	Good	Sufficient Coins
	Low	Good	Coins Low
	Empty	Fatal	Out of Coins
	Missing	Fatal	Out of Coins

Configuring Communications

The default registry settings for communications can be modified to configure requirements that cannot be configured through the Supervisor Configure menu on the SST.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Service Class

The following registry key contains the setting for the service class:

```
HKLM\SOFTWARE\NCR\Advance NDC\VPIComms\VPICommsServiceProgID
```

The default setting is for TCP/IP, as follow:

```
NCRaVPITCPIP.TCPIPCommsService
```

For SSTs using a PCCM-based protocol, you must change the setting to the following:

```
NCRaVPIPCCM.PCCMCommsService
```

For details of PCCM, refer to the following:

- *APTRA Communications Feature, User's Guide*
- APTRA On-line Documentation, CCM PCCM.

For SSTs using CCM VISA2, you must change the registry setting to the following:

```
NCRaVPIVISA2.VISA2CommsService
```

Clearing the Communications Buffer

On power up or following a communications recovery, the buffer can be cleared to prevent the host receiving confusing responses due to message buffering. Whether the buffer is cleared is set using the following registry key:

```
HKLM\SOFTWARE\NCR\Advance NDC\VPIComms\ClearCommsBuffer
```

Valid values are as follows:

- 0 to clear the communications buffer. This is the default.
- 1 to retain the messages in the communications buffer.

Off-line Timer

When off-line mode is entered, Advance NDC checks every five seconds whether communications have been re-established by the communication protocol. If communications are re-established this timer (Timer 06) is cancelled. If communications are not re-established before this timer expires, the communication link is closed for 10 seconds, then re-opened. This timer and the five second test of the communication state are then restarted.

To configure the number of seconds before timer expiry, edit the following registry key:

```
HKLM\SOFTWARE\NCR\Advance NDC\VPIComms\OffLineTimer
```

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

The default value is 600 seconds and the valid range of values is 0-99999.

TCP/IP Configuration

As well as configuring TCP/IP through Supervisor menu options, you can edit the registry to modify the Keep Alive values, or the TCP/IP configuration file to specify a domain name server (DNS).

Domain Name Server If you are using CCM TCPIP and want to configure a domain name server (DNS), you must edit the TCP/IP configuration file, *TCPIPCommsServiceconfig.xml*, directly and add the name of the DNS to the `RemoteHost` attribute.

Caution If you edit the TCP/IP configuration file to add a DNS attribute and subsequently the 0 RMT ADDRESS option in Supervisor is used to update the TCP/IP settings on the SST, your changes will be overwritten.

For details of the "<TCPIPCommsLink>" element, refer to the *APTRA On-line Documentation, CCM TCPIP*.

Keep Alive This option can be enabled or disabled using the Advance TCP/IP Config menu. The Keep Alive option is enabled by default. When enabled, the Keep Alive settings are controlled by the following registry key:

```
HKLM\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters
```

The values are in milliseconds for the following:

```
KeepAliveInterval
KeepAliveTime
```

Dialup Configuration

Dialup communications is enabled with the following registry setting:

```
HKLM\SOFTWARE\NCR\Advance NDC\VPIComms\VPICommsServiceProgID
```

```
NCRaVPIVISA2.VISA2CommsService
```

Configuration is required for both lower-level (modem) communications and the application. Configuration values are held as follows:

- For the application parameters, under the following registry key:

```
HKLM\SOFTWARE\NCR\Advance NDC\VPIComms\dialup
```

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and/or distribution is strictly prohibited.

- For the protocol and modem parameters, in the CCM VISA2 XML file, *VISA2CommunicationServiceconfig.xml*. This file has default values only and has not been preconfigured.

Status Message Suppression When a cassette reaches the low threshold, an unsolicited cash handler device status message is sent to Central. This message is sent each time the cassette is used until the cassette either runs out of notes or is replenished. In some dialup environments, these repeated messages can impact the network by requiring additional dial-outs. As this can lead to additional network costs and reduction in the cost advantages of dialup communications, Advance NDC has introduced a method of suppressing these messages.

The number of messages to send for each cassette before suppressing any further messages is set using Application Parameters under 38 - DIALUP on the Configure menu in Supervisor. This updates the following registry key:

```
HKLM\SOFTWARE\NCR\Advance NDC\VPIComms\Dialup\SuppressCashLow
```

Valid values are as follows:

- 0 to disable message suppression. This is the default
- 1-9 to specify the number of messages sent for a cassette before suppression.

Alphaumeric Entry From release 3.01, a defined set of alphanumeric values and special characters can be configured for:

- Communications access routing information (BIN)
- Terminal Identification (TID) information
- Primary Number
- Secondary Number
- Modem Initialisation String
- Network Address String.

The values can also be configured by any of the following methods:

- Preconfigured and deployed at rollout
- Edited directly on a development system
- Modified on a secure system using the Supervisor Copy Off / Copy On function.

Preconfiguration If editing the CCM VISA2 XML file for preconfiguration, the XML restrictions described below must be applied. Any preconfiguration entry must have a `CommsLinkID` of `TPA CONNECTION` and there must only be one `VISA2CommunicationsLink` element in the file.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Connect Life Time On this menu, the two options have the following meaning:

- Normal = in transaction
- Pre-Dial = in session.

Copy On/Off To facilitate configuration, or modification that is unique to a particular SST, an option is provided in Supervisor to transfer the XML configuration file or the application registry settings, in XML format, to and from diskette. This functionality can be used, for example, to configure the SST at start of day.

The function does not validate any changes made to the XML file. It is responsibility of the person making the change to ensure the validity both of syntax and values.

XML Restrictions Certain characters are treated by XML as special characters. For example, an ampersand (&) is a XML special character and must be entered using the ASCII decimal value (38) as & in XML.

NCR recommends that you have a knowledge of XML before editing the CCM VISA2 XML file, or use an XML editor.

Diagnostics For dialup diagnostics to work correctly, configuration is required, either through preconfiguration or through diskette transfer.

The screens for dialup diagnostics are not held in *resrvd.def* but in the forms supplied with CCM VISA2.

For details of the diagnostic tests available, refer to the *APTRA Advance NDC, Supervisor's Guide.*

Note: Localisation of these forms is not supported.

Dialup Timers and Modem Baud Rate

Except for Timer 3, which is sent from Central, the timers and baud rate are configurable for numeric entry through the Dialup Config menu in Supervisor. The duration of the application timers is set in accordance with the settings for the modem timers.

Setting the timers to the correct value is important for dialup communications to function correctly. For details of the timers and the calculations to use when determining the length of time to set, refer to Appendix O, "CCM VISA2 Dialup System" in the *APTRA Advance NDC, Reference Manual.*

Activity Timer This timer should be set to a large value, determined by how often the host is expected to send messages in quiet periods. The default value is 3000 seconds.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Error Redial Timer This timer specifies how long the application will wait before it tries dialling again to establish a connection after an error has been experienced with a previous dial attempt. This timer should be set to a shorter value than the Activity Timer. The default is 300 seconds.

Timer 3 The duration of this timer (communications response timer, downloaded from the host) has to be sufficient to allow the dialup connection to complete its operations. In non-dialup systems, Timer 3 typically has a value of 20 to 30 seconds. In a dialup system the typical value is 40 to 60 seconds.

Modem Connect Timer The Modem Connect Timer determines how long CCM VISA2 will wait for an attempt to connect to the host before signalling an error. The duration of this timer must be shorter than Timer 3.

Dial on Send Timer The duration of this timer must be at least as long as the time taken to transmit the longest message (for example, a Screens or State message) and receive the corresponding reply.

For EJ upload, the Dial on Send timer determines the amount of time the host has available to send an acknowledgement to an EJ upload message. The call will be disconnected after the Dial on Send timer expires.

Status Handling

As devices are accessed through the CEN-XFS interface, the status information from XFS is mapped to the corresponding Advance NDC status.

Printing Configuration

Reconfiguration of the default settings provided by Advance NDC may be required for particular printers.

Receipt Printer

The default configuration for the receipt printer service provider is the following:

- Thermal printer
- Variable length receipt
- 'Black Mark' disabled, to prevent the printer automatically ejecting receipts that exceed the printer's standard receipt length.

Note: Any receipts that are not removed will be retracted when the hardware has the capability on the following:

- Start of Day

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

- Close state
- INIT RECEPT.

Statement Printer

You can set a maximum statement length, which sets the maximum number of lines to be printed before a cut. This ensures that the statement will be cut when the black mark is not used. The registry key is under:

```
HKLM\SOFTWARE\NCR\Advance NDC\PRINTING\STATEMENT\Length
```

The default value is `20`. Valid values are between `12` and `92`, inclusive.

Form-Based Printing The Open Statement Printer can now be used where forms are dynamically generated for each print operation. The form generation process can be customised using the registry keys described in Table 5-4.

Table 5-4
Form-Based Printing Registry Keys

Registry Key	Description
HKLM\SOFTWARE\NCR\Advance NDC\PRINTING\STATEMENT FORMS\Dynamic Forms *See* Table Note 1	Specifies whether forms can be used. Valid values are as follows: ● 0 to prevent form-based printing ● Any non-zero number to allow form-based printing
HKLM\SOFTWARE\NCR\Advance NDC\PRINTING\STATEMENT FORMS\Media Definition	Specifies the media definition name to use when printing forms. A media definition file must be present on the system, and contain the definition matching the name used in this registry key. Valid values are as follows: ● Blank. This is the default. However, NCR printer service providers require the media name to be defined or the print will fail. ● The name of the definition. By default, this is set to Statement80
HKLM\SOFTWARE\NCR\Advance NDC\PRINTING\STATEMENT FORMS\Function Names	Specifies the function name to use for field conversion during form generation. The only valid value is as follows: ● ConvertStatementForms. This is the default.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Registry Key	Description
HKLM\SOFTWARE\NCR\Advance NDC\PRINTING\STATEMENT FORMS\Print Data\Form Units	Specifies the units of measurement used in forms generation. The only valid value is as follows: ● ROWCOLUMN, 1, 1. This is the default.
HKLM\SOFTWARE\NCR\Advance NDC\PRINTING\STATEMENT FORMS\Print Data\Alignment	Specifies the starting co-ordinates for the form using ROWCOLUMN format. Valid values are vendor dependent, but will be one of the following: ● TOPLEFT, 1, 1. This is the default. ● TOPLEFT, 0, 0
HKLM\SOFTWARE\NCR\Advance NDC\PRINTING\STATEMENT FORMS\Print Data\Forms Extension	Specifies the file extension for the generated form. The value is found in the key values for your printer. By default, it is .DEF.
HKLM\SOFTWARE\NCR\Advance NDC\PRINTING\STATEMENT FORMS\Print Data\Forms Name	Used to generate a unique form name, and provide the filename of the generated form. The value is found in the key values for your printer.
HKLM\SOFTWARE\NCR\Advance NDC\PRINTING\STATEMENT FORMS\Print Data\Forms Path	Specifies the directory in which to generate the form. Some service providers need the form to be stored in a specific directory, or the form will not load at run time. Refer to the appropriate vendor publications for the value required for this key. The default paths are as follows: ● `C:\Program Files\NCR APTRA\Advance NDC\Printing\Statement\Forms` ● `C:\Program Files\NCR APTRA\Advance NDC\Printing\Statement\Media`
HKLM\SOFTWARE\NCR\Advance NDC\PRINTING\STATEMENT FORMS\Print Data\Font Identifier*<var>*\Font Name *See* Table Note 2	The name of the font, used in the font field of the form. This is vendor-dependent. Refer to the appropriate vendor publications for the values accepted for this key. By default this key is blank, which means that the default font will be used.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Registry Key	Description
HKLM\SOFTWARE\NCR\Advance NDC\PRINTING\STATEMENT FORMS\Print Data\Font Identifier\<*var*>\CPI *See* Table Note 2	The characters per inch setting, used in the CPI field of the form. This is vendor-dependent. Refer to the appropriate vendor publications for the values accepted for this key. The default is 10 CPI.
HKLM\SOFTWARE\NCR\Advance NDC\PRINTING\STATEMENT FORMS\Print Data\<*var*>\Font Size *See* Table Note 2	Specifies the font size in characters. Valid values are as follows: ● 1 for single size ● 2 for double size ● 3 for condensed size
HKLM\SOFTWARE\NCR\Advance NDC\PRINTING\STATEMENT FORMS\Print Data\<*var*>\Font Style *See* Table Note 2 and Table Note 3	Specifies the font style. Valid values are as follows: ● NORMAL. This is the default ● BOLD ● ITALIC ● UNDER ● DOUBLEUNDER ● DOUBLE ● TRIPLE ● QUADTRIPLE ● STRIKETHROUGH ● ROTATE90 ● ROTATE270 ● UPSIDEDOWN ● PROPORTIONAL ● DOUBLEHIGH ● TRIPLEHIGH ● QUADRUPLEHIGH ● CONDENSED ● SUPERSCRIPT ● SUBSCRIPT ● OVERSCORE ● LETTERQUALITY ● NEARLETTERQUALITY ● DOUBLESTRIK ● OPAQUE

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Registry Key	Description
HKLM\SOFTWARE\NCR\Advance NDC\PRINTING\STATEMENT FORMS\Print Data\LPI Index\<*var*>\LPI *See* Table Note 4	The lines per inch setting, used in the LPI field of the form.

Table Note 1: If using forms-based printing with a Personas class SST, you must update this field as follows:

- Update the `Dynamic Forms` setting in the *PersonasPrinters.reg* file from 0 to 1.
- Run the modified file as a custom action at the end of the Advance NDC runtime installation.

Table Note 2: The <*var*> part of these registry keys represents a font identifier. Any ASCII character can be used, however, only those present in the print stream are actioned.

Table Note 3: Values can be ORed together using the "|", for example, DOUBLE | ITALIC. Some styles may be mutually exclusive or combine to provide unexpected results.

Table Note 4: The <*var*> part of this registry key represents the lines per inch indexes, 0 to 7 inclusive.

USB Receipt and USB Journal Printers

For details of the differences in registry settings for the USB Receipt and Journal printers, refer to the *APTRA Advance NDC, Reference Manual.*

See the APTRA on-line documentation for details of the printer SP.

The following registry settings are used:

Table 5-5
USB Receipt and Journal Registry Settings

Registry Key	Description
CHAR_MAP_FILENAME	Stores the character map filename, which translates the code page used by the USB Receipt and Journal printer into the corresponding Arabic and International characters.
CHAR_MAP_DESIGNATOR	Stores the designator used to identify the user-specified font in the mapping file.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

For details of the mapping file and associated registry keys, refer to the *APTRA Advance NDC, Reference Manua*.

Creating *.prn* Files for Use With USB Receipt and USB Journal Printers *.prn* files for the SDC printers will not print on the USB printers. To create *.prn* files for use with USB Receipt and USB journal printers, do the following:

1. Locate the *NCR80RDR.INF* file on your PC and note the path. This file is installed as part of an APTRA XFS development installation.

2. Select **Start** from the Windows Taskbar, then **Printers and Faxes | Add Printer**. The Add Printer Wizard opens.

3. Select the **Next** button.

4. Select the **Local printer attached to this computer** radio button and deselect the **Automatically detect and install my Plug and Play printer** check box. Select the **Next** button.

5. Select the **Use the following port** radio button, and select **File (Print to File)** from the associated drop-down menu. Select the **Next** button.

6. Select the **Have Disk** button. The Install from Disk dialog box appears, select the **Browse** button and browse to the *NCR80RDR.INF* file in the location you identified in step 1. Select the *NCR80RDR.INF* file from the file list, and select the **Open** button. You are returned to the Install from Disk dialog box, select the **OK** button to return to the Add Printer Wizard.

7. Select "NCR USB 80mm Renderer" from the **Manufacturer** list box, and then select the **Next** button.

8. Choose the default printer by selecting one of the following radio buttons:
   - If you have a different printer that you want to use as the default, select the **No** radio button
   - If you want to use the NCR USB 80mm Renderer as the default, select the **Yes** radio button.

   Select the **Next** button.

9. Select the **Do not share this printer** radio button, then select the **Next** button.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

10 As this is printing to a file, you do not need to print a test page, so select the **No** radio button, then select the **Next** button.

11 Select the **Finish** button.

As the NCR USB 80mm Renderer driver is not signed, you might be prompted before installation. Choose to continue with the installation. A message will appear notifying you of a successful installation.

To produce a *.prn* file, open the graphic you require and select Print to File.

Non-Thermal Printers

Reconfiguration is required for journal, receipt and statement printers that do not support the printer control codes set as default in Advance NDC, such as ESC G for graphics. If an unsupported control code is sent for printing, a square bracket ([ ) is printed.

The configuration can be made either through the host or through the registry settings on the SST. The registry keys for printers are under:

```
HKLM\SOFTWARE\NCR\Advance NDC\PRINTING
```

The following subkeys may need to be reconfigured for your particular printer:

- BARCODE
- GRAPHICS
- HZ_HGHT_BARCODE
- POS_HRI_CHAR
- SW_ON
- SW_OFF
- WIDTH_BARCODE

For information about supported control codes, refer to the documentation for your particular printer.

Promote Coupons

The format to use when printing Promote coupons has to be configured for the specific receipt printer in use.

To permit Promote coupon printing, create a string value registry key called `PROMOTE_COUPON_FORMAT` under:

```
HKLM\SOFTWARE\NCR\Advance NDC\PRINTING\Receipt\
```

Valid values are as follows:

- `RCPT:PFMHIGHRES` for NCR 40 column thermal printers

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

- `RCPT:PFMLOWRES` for NCR 60mm thermal printers
- `RCPT:BMP-HIGHRES` for NCR USB printers
- `RCPT:BMP-LOWRES` for non-NCR printers

Print Track 2 to Journal

If enhanced configuration parameter option 37 is set to 001, the first 22 characters of track 2 are printed to the journal, or electronic journal with the defined masking applied. This is done between reading the card track data and performing the FIT search.

The characters to mask are defined by the first line of screen t15. The following rules are applied to the mask:

- Journal any character that is represented by a zero (‘0’) in the screen. For example, enter the following in screen t14:

  ```
  CARD: 00000000000000000000
  ```

  To journal the whole of the track 2 journal, as follows:

  ```
  CARD: ;1234567890123456789=2
  ```

- Journal any character that is represented by an ‘X’ in the screen as ‘X’, masking that character. For example, enter the following in screen t14:

  ```
  CARD: 0000XXXX0000XXXX0000XX
  ```

  To mask the equivalent characters with an ‘X’ on the journal, as follows:

  ```
  CARD: ;123XXXX8901XXXX6789XX
  ```

- Insert any character that is represented by anything other than a ‘0’ or an ‘X’ into the journal. For example, enter the following in screen t14:

  ```
  CARD: 0000PPPP00000000000000
  ```

  To journal the following characters:

  ```
  CARD: ;123PPPP45678901234567
  ```

- Only the characters represented by the mask will be journalled. For example, enter the following in screen t14:

  ```
  CARD: 0XXP0
  ```

  To journal only the following characters:

  ```
  CARD: ;XXP3
  ```

- Only up to 22 characters will be journalled. For example, enter the following in screen t14:

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

```
CARD: 0000000000XXXXXXXXXXXXXXXX
```

To journal 22 characters, as follows:

```
CARD: ;123456789XXXXXXXXXXXX
```

For further information on option 37, refer to the *APTRA Advance NDC, Reference Manual*.

Cheque Processing Module

The location of an image lifted by the cheque processing module (CPM) is set using the following registry key:

```
HKLM\SOFTWARE\NCR\Advance NDC\CPM
```

The default path is `C:\Program Files\NCR APTRA\Advance NDC\Data`.

If your printer can use the Epson Graphics format, you can print the lifted image. The following registry keys provide the escape sequence to allow printing of Epson Graphics:

```
HKLM\SOFTWARE\NCR\Advance NDC\PRINTING\Receipt\CHEQUE

HKLM\SOFTWARE\NCR\Advance NDC\PRINTING\Journal\CHEQUE
```

Note: The journal printer registry setting is left empty to avoid the inclusion of bitmap data in the EJ.

If the value is empty, the escape sequence to print a cheque image is ignored. For details of escape codes, refer to the *APTRA Advance NDC, Reference Manual* or your printer's documentation.

Front Keyboard

The front keyboard or the PIN SP must be configured so that, as well as all numeric keys (0-9), the ENTER, CLEAR and CANCEL keys are set up. Default key configurations are provided, however, NCR recommends the CLEAR key be configured to clear the full input buffer, not just one character.To support this the CLEAR key must return the WFS_PIN_FK_CLEAR value to the application.

To do this for NCR SSTs follow these instructions for the Aggregate Builder Tool:

1. On the development PC, import the APTRA XFS Platform Aggregate from the distribution media.
2. Create a new archive on the development PC by selecting a new empty directory.
3. In the Aggregate Builder tool, open the APTRA Self Service Support Aggregate, select the version and click Contents.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

4 Scroll to the Keyboards component and select the current revision, for example 04.00.03. In the Configuration Sets window at the bottom, it will show the Default (NCR) set.

5 Select the Default (NCR) set with right mouse and perform Duplicate.

6 Rename the newly created set, for example to Advance NDC.

7 Right click the new set and select open, click Change. This brings up the Change Wizard with the default keyboard selected.

8 Click Copy to bring up a Custom Keyboard Information dialog. Pick a Custom keyboard feature number from the dropdown menu.

9 Enter a description in the Custom keyboard description dialog, for example, "APTRA NDC Keyboard".

10 Now click Keyboard to configure the Custom keyboard for the Front Keyboard.

11 Now find the Backspace key and double click it to bring up the Set Key Configuration dialog.

12 In the list of Virtual key names find Delete (= Clear) and select it. Click OK twice to close both dialogs.

13 Click Next to proceed to Step 2 of the wizard detailed below under "Operator Keyboard".

The function display keys (FDKs), if present, must be set up with the following layout:

Table 5-6
Front Keyboard Layout

WFS_PIN_FK_FDK01		WFS_PIN_FK_FDK05
WFS_PIN_FK_FDK02	*front*	WFS_PIN_FK_FDK06
WFS_PIN_FK_FDK03	*display*	WFS_PIN_FK_FDK07
WFS_PIN_FK_FDK04		WFS_PIN_FK_FDK08

Operator Keyboard

The operator panel keyboard or TTU SP are configured by default, supporting FDKs as individual keys. However, NCR recommends the CLEAR key be configured to clear the full input buffer, not just one character.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

To configure this for an NCR SST follow these instructions as "Step 2 of 2" of the Change Wizard as introduced in section "Front Keyboard":

1. Click Copy to bring up a Custom Keyboard Information dialog. Pick a Custom keyboard feature number from the dropdown menu.

2. Enter a description in the Custom keyboard description dialog, for example, “APTRA NDC Rear Keyboard”.

3. Now click Keyboard to configure the Custom keyboard for the Rear Keyboard.

4. Now find the Backspace key and double click it to bring up the Set Key Configuration dialog.

5. In the list of Virtual key names find Delete (= Clear) and select it. Click OK to close the dialog.

6. Click Finish to close the wizard. Click OK to close the Configuration Property sheet.

7. Now click Profiles under the APTRA Self-Service Support aggregate to create a new profile. Name the new profile for example, SSSProfile.

8. Select the profile and scroll to the Keyboards component. Under the Installation Instruction dropdown select the configuration set AdvanceNDC that has been created.

9. Now click Profiles under the APTRA XFS aggregate to create a new profile. Name the new profile for example, XFSProfile.

10. Select the profile and scroll to the APTRA Self-Service Support aggregate. Under the Installation Instruction dropdown select the profile SSSProfile that has been created.

11. Export the APTRA XFS Platform Aggregate by selecting the aggregate version, right-clicking and selecting Export from the context menu.

12. Close the Aggregate Builder tool.

13. The customised aggregate which has now had a custom keyboard configuration set can be deployed to the SST network.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Suspend Timeout

As devices are now accessed through the CEN-XFS interface, clearing the Suspend state is controlled by the SPs, but the CEN-XFS specification does not define whether applications can clear Suspend states themselves.

The Advance NDC application detects a Suspend state at the end of the Close state and before the Idle state recommences. The SP clears the Suspend state at the SP level, after the timeout defined in the SP configuration.

When the SST detects that a device has been tampered with, the application goes out of service temporarily and the SST goes into Suspend mode until one of the following occurs:

- The host sends a go-in-service (GIS) message
- The host sends a out-of-service (OOS) message
- The application Suspend timeout expires
- Supervisor is entered by pressing the Mode switch.

The SST does not go into Suspend mode until the end of the transaction in which tampering has been detected.

When the SST attempts to come out of Suspend mode, the SP attempts to clear the device that caused the suspension of service. If this succeeds then the SST is no longer suspended.

The SPs have their own configurable timers for coming out of Suspend mode and will attempt to clear the devices when these timers expire. SPs report devices as still in the suspend state (`WFS_STAT_DEVUSERERROR`) until the SP timers expire. Registry keys are used to set the SP timers.

Although the host removes the application from the Suspend state, a command to change the mode (Supervisor, In Service, Out of Service, Initialise or Power-up) will not be actioned nor a Ready 9 response sent until the SP has cleared the device Suspend state. When the Suspend state is cleared, Advance NDC responds to the terminal command and then sequentially processes all other messages received.

Setting the SP Timeout

The Suspend Timeout registry key is located in the `GENERAL_CONFIGS` subkey for each supported SP. The default timeout duration for Advance NDC is 120 seconds, which is the minimum duration recommended in Advance NDC. For more information about setting the Suspend Timeout registry key, refer to the APTRA on-line documentation, under **APTRA XFS | Programmers Reference | XFS Service Providers**.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Set the SP timers to a value greater than the time it takes to complete a transaction. If the value is less, the application will not enter the Suspend state even though the host receives a Suspend message. This is the equivalent of the application going into a Suspend state and coming out again after a very short application timeout.

Note: The GBRU SP Suspend Timeout is set by Advance NDC to the maximum limit of 900 seconds. This means that Advance NDC can issue the reset command to bring the GBRU out of suspend in any of the following situations:

- A Go In Service or Out Of Service message is sent
- The timeout for Advance NDC expires when the GBRU is in a suspend condition
- Supervisor mode is entered or exited.

For more information, see "GBXX Cassette Configuration" on page 5-47.

For more information about Suspend time outs, refer to the *Extensions for Financial Services (XFS) interface specification (CWA 14050).*

Setting the Application Timeout

The Advance NDC application timeout is set to 5 minutes (300 seconds). The timer can be changed through the Author (Timer worker "5 Mins"). The application timeout must always be longer than that of the service provider (SP). This will allow the SP to clear the device ready for use by the application. If the application timer setting is less than the SP timer, the application will stay in the Suspend state until the SP timer expires. Moreover, if the host sends a GIS or OOS message, the SST will stay in Suspend mode until the SP timer expires.

Service Providers

The Advance NDC installation configures the NCR service providers (SPs) to ensure they will function with Advance NDC.

For SP configuration on other vendors' SSTs, refer to the vendor-specific documentation and the *APTRA Advance NDC, Multi-Vendor Support Reference Manual.*

Service Provider Reset Value

By default, only one attempt can be made to clear a device fault before intervention through the System Application or other Vendor Dependent Mode (VDM) is required. The following examples illustrate how this can affect Advance NDC:

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

- The Service Provider (SP) has previously tried and failed to reset a BNA device, the INIT BNA operation also fails unless the System Application is used first to clear the device
- The reset command sent on Supervisor exit is not successful if a previous attempt to reset a device has been made and the System Application has not been used to clear the device in the meantime.

If a reset fails, use the System Application to clear a device.

Uninterruptible Power Supply

UPS devices are supported through the SIU service provider (SIU SP) and can be configured through the System Application. For full configuration details refer to the APTRA on-line documentation under **APTRA XFS | Programmer's Reference | Feature Management | Power Supplies | PCUnintPowerSupply**. See Table 5-7 for a summary of the UPS power management.

Table 5-7
UPS Power Management

Mains Failed	Battery Low	Action taken
False	False	SST runs as normal using the mains supply
False	True	SST switches to Out of Service until the battery level exceeds the low threshold, at which point it goes in service. If the mains power fails again during this period, SST performs a controlled shutdown.
True	False	SST switches to Out of Service
True	True	SST performs a controlled shutdown

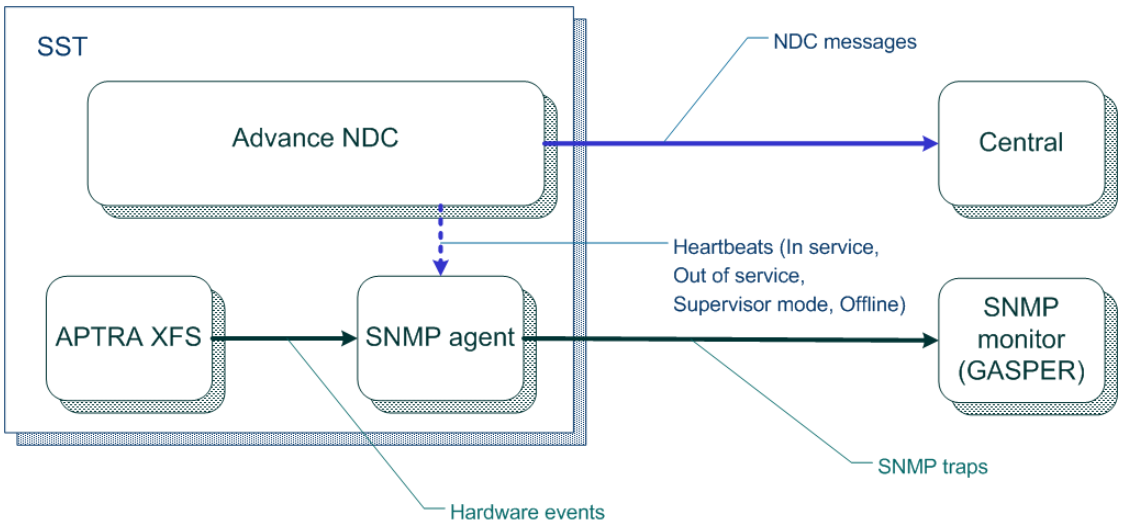
Simple Network Management Protocol

Advance NDC supports the sending of application-related SNMP traps. When changes in state occur in Advance NDC, events (known as traps) are sent by the SNMP agent to the SNMP manager. Only the traps detailed here originate in Advance NDC. All other traps, for events such as hardware state changes, originate in APTRA XFS. See Chapter 2 of the SNMP Agent, Reference Manual for further information.

The ideal communications flow is illustrated in Figure 5-1 below.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Figure 5-1
APTRA and SNMP



Advance NDC requires aTCP/IP connection to the SNMP server to send SNMP traps. There are two types of trap used, heartbeats and mode changes: see "Application Heartbeats" on page 5-30 and "Advance NDC Mode Changes" on page 5-31.

There are two methods of implementing this functionality in Advance NDC, as follows:

- As recommended by the *APTRA SNMP Agent User Guide* (NCR SNMP): see "NCR SNMP Implementation" on page 5-31
- GASPER-compatible: see "GASPER-Compatible Implementation" on page 5-32

Application Heartbeats

Regular signals (heartbeats) from the Customisation Layer of Advance NDC allow Central to monitor the communications connections.

By default, the initial heartbeat is generated at Start of Day with subsequent heartbeats generated every 30 minutes. These timings can be changed by editing the `HKEY_LOCAL_MACHINE\SOFTWARE\NCR\Advance NDC\SNMP\HeartBeatDelay` registry key. As there is no dependency on other signals, you can choose to reduce or increase the period between heartbeats.

The value for this registry key is given in seconds. The default is 1800 seconds, or 30 minutes.

NCR SNMP implementation uses the Generate Heartbeat type trap.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and/or distribution is strictly prohibited.

GASPER-compatible implementation uses the Report State type trap: see Table 5-10 for further details.

Advance NDC Mode Changes

Advance NDC mode changes generate traps to report application events that require logging or possible action by the SNMP manager.

The NCR SNMP implementation uses a pair of traps listed in tables "NCR SNMP Service Traps" and "NCR SNMP Mode Change Traps" on page 5-32.

The GASPER-compatible implementation uses only the traps listed in table "GASPER-Compatible SNMP Traps" on page 5-32.

NCR SNMP Implementation

To use this implementation:

- Set the `HKEY_LOCAL_MACHINE\SOFTWARE\NCR\ Advance NDC\SNMP\NCRSNMP` registry key to 1.
- This implementation sends two types of traps:
  - — one communicates whether the SST is available for use, that is whether it is In or Out of service. This is the Consumer Application Status Trap. See table "NCR SNMP Service Traps" on page 5-31 for details.
  - — a second identifies the mode change. For example, when the first trap has identified that the SST has gone out of service, this trap identifies whether it is Out of service, Offline, or in Supervisor mode. This is the Application Event trap. See table "NCR SNMP Mode Change Traps" on page 5-32 for details.

Table 5-8
NCR SNMP Service Traps

Severity (Enum)	Meaning within Advance NDC
0	NDC is in service
2	Supervisor mode has been entered, NDC is either out of service or in offline mode

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Table 5-9
NCR SNMP Mode Change Traps

Application Name (String)	Event (String)	Severity (Enum)	Meaning within Advance NDC
"AA-NDC"	"IN_SERVICE"	0	Normal operation - In service
"AA-NDC"	"OUT_OF_SERVICE"	2	Critical - Out of service
"AA-NDC"	"OFFLINE_MODE"	2	Critical - Offline
"AA-NDC"	"SUPERVISOR_MODE"	2	Critical - Supervisor
"AA-NDC"	"SUSPEND_MODE"	2	Critical - Suspend condition
"AA-NDC"	"UnKnown"	1	Warning - State Unknown

GASPER-Compatible Implementation

To use this implementation:

- Set the `HKEY_LOCAL_MACHINE\SOFTWARE\NCR\Advance NDC\SNMP\NCRSNMP` registry key to 0.

  Note: This is the default setting for this registry key

- The Application heartbeat and mode change traps are given in Table 5-10.

Table 5-10
GASPER-Compatible SNMP Traps

Name (String)	Data (String)	Alert Hint	Meaning within Advance NDC
"APTRA WEBATM STATUS"	"1"	TRUE	In Service
"APTRA WEBATM STATUS"	"2"	TRUE	Out of Service
"APTRA WEBATM STATUS"	"3"	TRUE	Supervisor
"APTRA WEBATM STATUS"	"4"	TRUE	Exit Supervisor
"APTRA WEBATM STATUS"	"5"	TRUE	Suspend
"APTRA WEBATM STATUS"	"I"	TRUE	Offline
"APTRA WEBATM STATUS"	"J"	TRUE	Unknown
"APTRA WEBATM HEARTBEAT"	DD/MM/YYYY hh:mm:ss (current date and time)	TRUE	Heartbeat

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and/or distribution is strictly prohibited.

Settlements Screen Customisation

In previous releases of Advance NDC, the Settlements supervisor transaction screens were offered through fixed text display.

This release now allows you to configure these screens.

Changing these screens may overwrite the input data, or result in misalignment. If this happens, update the Operator Echo workers in the Application Core to reposition the input data. The affected screens and associated workers are shown in Table 5-11.

Table 5-11
Correcting Settlement Screen Customisation Misalignment

Screen Number	Operator Echo Worker Name
i93	One Key Input: 1
i94	32 Key Input
i95	
i98	8 Key Input
	12 Key Input

For further information on the screens, refer to the *APTRA Advance NDC, Reference Manual*. For further information on the workers, refer to the Author on-line help.

AVI System Limit

There is an operating system imposed system limit on the number of distinct AVI windows open within a process. This means that you cannot use more than 24 AVI files.

If you need to play more than 24 AVI files, use partial AVI files. A partial AVI is not a full screen display, but is superimposed on a JPEG or PCX background. The JPEG or PCX can provide a static picture over which the partial AVI files can provide the moving part of the display. This separation of the static and moving parts helps to reduce the number of AVI files required.

Note: The 24-file limit also applies to partial AVI files.

Cardless Transactions

Instead of using a card, cardholder transactions can be started using FDK, FDK emulation, or PIN pad key entry. If an active FDK or PIN pad key is pressed with no card inserted, the transaction proceeds as configured in the registry. If a card is inserted and a key pressed simultaneously, the card takes priority.

The registry key used to configure a cardless transaction is:

```
HKLM\SOFTWARE\NCR\Advance NDC\Extensions\Cardless Transaction
```

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

As this option can only be set up through the registry and not through the enhanced configuration parameters load message, all details are included here.

Table 5-12 describes the registry settings used to define the processing of cardless transactions.

Table 5-12
Cardless Transaction Registry Settings

Registry Value	Description
`Next State Number`	The initial state number of a cardless state. Default is 000, cardless transactions not active. This is configurable using option 77 in the enhanced configuration parameters load.
`SkipFitSearch`	Defines whether a FIT search is carried out based on Track 2 data. Default is 0, FIT search is carried out.
`Track 2 Data`	Track 2 data for cardless transactions.
`Active FDK Mask`	FDK mask value for cardless transactions. Default is 0. If this is not set, the default is used.
`Active PIN Pad Key Mask`	Numeric mask value for cardless transactions. Default is 0. If this is not set, the default is used.

Note: If both `Active FDK Mask` and `Active PIN Pad Key Mask` are set to 0, no keys are activated and only a card-initiated transaction is allowed.

If the defined FDK, touch area, or numeric is pressed, and no card is inserted, and the FIT search is not to be skipped, the Track 2 Data registry is checked and a FIT match performed.

Note: If no Track 2 Data is present but a FIT search is to be skipped, no FIT match is performed.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

If the FIT matches or the FIT search is to be skipped, the next state number is obtained from the Next State Number registry key and processing continues as normal.

If the FIT does not match, the standard close state processing using the next state from the card entry state is used.

Key Mask Definition

The `Active FDK Mask` and `Active PIN Pad Key Mask` registry settings define the FDK and PIN pad keys to activate.

These masks are binary-coded values. To activate an FDK, PIN pad key or touch area, set the associated bit, as shown in Table 5-13. To activate more than one FDK or PIN pad key, XOR the codes.

Table 5-13
Key Mask Settings

Set Bit	Associated FDK	Associated PIN Pad Key
0	A	0
1	B	1
2	C	2
3	D	3
4	F	4
5	G	5
6	H	6
7	I	7
8		8
9		9
10		ENTER

For example, if you wanted to set the ENTER, 8, 5, and 2 PIN pad keys, you would enter 1316 in the `Active PIN Pad Key Mask` registry value. This is calculated as shown in Table 5-14.

Table 5-14
Binary to Decimal Conversion Example

Key	ENTER	9	8	7	6	5	4	3	2	1	0
Binary	1	0	1	0	0	1	0	0	1	0	0
Decimal	1024	512	256	128	64	32	16	8	4	2	1
Active	1024		256			32			4		
Total	1316										

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Enhanced EJ Backup

From release 3.01, Advance NDC provides the ability to create multiple EJ backup files. Multiple EJ backup allows up to 1000 EJ files to be stored depending on disk space. Standard EJ backup is the default, allowing only a single EJ backup file to be retained.

EJ backup can be changed either through option 36 of the Enhanced Configuration Parameters Download, or through Supervisor.

Note: There is an option in Supervisor to prevent the host from changing this setting. For more details, see "Disabling Host Control of Enhanced EJ Backup" on page 5-41.

Each method updates the following registry key:

```
HKLM\SOFTWARE\NCR\Advance NDC\EJOptions\EJMode
```

Valid values for this key are as follows:

- 0 for standard EJ backup. This is the default
- 1 for multiple EJ backup.

If standard EJ backup is enabled, files are backed up to `C:\Program Files\Advance NDC\data`.

If multiple EJ back up is enabled, files are backed up to `C:\Program Files\Advance NDC\data\ejbackups`.

Maximum Number of Backups

With multiple EJ backup set, the maximum number of backups can be set using the Configure menu in Supervisor mode.

This updates the following registry key:

```
HKLM\SOFTWARE\NCR\Advance NDC\EJOptions\NoOfBackups
```

Valid values for this key are `2` to `1000`, provided there is enough space on disk. The default is `10`.

If the requested number of backups exceeds the available disk space, the following message is displayed:

`INVALID; MAX POSSIBLE VALUE = <var>`

Multiple Destinations for EJ Backup

Advance NDC can be configured to prompt for a drive each time the EJ is backed up or copied using the Configure menu in Supervisor mode.

This sets the following registry key:

```
HKLM\SOFTWARE\NCR\Advance NDC\EJOptions\MultipleDestination
```

Valid values for this key are as follows:

- 0 for single EJ backup destination. This is the default.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

- 1 for multiple EJ backup destinations.

By default, if the CDRW is selected as the destination drive, the CD is ejected after the backup is taken. This can be controlled by the following registry key:

```
HKLM\SOFTWARE\NCR\Advance NDC\EJOptions\EjectCD
```

Valid values for this key are as follows:

- 0 to keep the CD in the drive following a backup.
- 1 to eject the CD following a backup. This is the default.

Automatic INIT Options

The INIT operation can now be run in a number of ways without having to manually select the INIT EJRNL Supervisor function. The backup file is created in the *c:\program files\advance ndc\data\EJBackups* directory. An optional copy of the automatic INIT file can also be saved to a specified destination.

For details of all options set in Supervisor mode, refer to the *APTRA Advance NDC, Supervisor's Guide*.

Note: Multiple destinations must be enabled to use any of the automatic INIT features.

Cutover

Advance NDC can be configured to run the automatic INIT whenever the EJ file reaches 90% of the specified maximum size using the Configure menu in Supervisor mode.

If Cutover is not used, an unsolicited status message is sent when the EJ file reaches 90% of the specified maximum size. This message is suppressed when Cutover is used. Also, when Cutover is used, the EJ file should never reach 100% of the specified maximum size, so the fatal message should never be seen.

For details of the unsolicited message, refer to the *APTRA Advance NDC, Reference Manual*.

This sets the following registry key:

```
HKLM\SOFTWARE\NCR\Advance NDC\EJOptions\EJCutover
```

Valid values for this key are as follows:

- 0 for disabled. This is the default.
- 1 for enabled.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Scheduled

Advance NDC can be configured to run the automatic INIT at a specified time and day or date using the Configure menu in Supervisor mode.

There is an option to enable or disable Scheduled automatic INIT. This sets the following registry key:

```
HKLM\SOFTWARE\NCR\Advance NDC\EJOptions\RecurrenceOptions\
EJRecurrence
```

Valid values for this key are as follows:

- 0 for disabled. This is the default.
- 1 for enabled.

Another option allows the setting of the interval at which to run the Scheduled automatic INIT.

The recurrence pattern sets the following registry key:

```
HKLM\SOFTWARE\NCR\Advance NDC\EJOptions\
RecurrenceOptions\EJRecurrencePattern
```

Valid values for this key are as follows:

- 1 for daily. This is the default.
- 2 for weekly.
- 3 for monthly.

The recurrence pattern selected determines the other options that must be set, as follows:

- Daily - set time
- Weekly - set day of the week and time
- Monthly - set day of the month and time.

The time sets the following registry key:

```
HKLM\SOFTWARE\NCR\Advance NDC\EJOptions\RecurrenceOptions\Time
```

Valid values for this key are any four digit number between 0000 and 2359 inclusive. This represents the 24 hour clock, and is based on the local time.

The day of the week sets the following registry key:

```
HKLM\SOFTWARE\NCR\Advance NDC\EJOptions\RecurrenceOptions\DOW
```

Valid values are as follows:

- 0 for Sunday. This is the default.
- 1 for Monday
- 2 for Tuesday

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

- 3 for Wednesday
- 4 for Thursday
- 5 for Friday
- 6 for Saturday.

The day of the month sets the following registry key:

```
HKLM\SOFTWARE\NCR\Advance NDC\EJOptions\RecurrenceOptions\DOM
```

Valid values for this key are in the range 1 to 31 inclusive. The default is 1 (the first day of the month). If the registry value is greater than the days in the current month, the automatic INIT will run on the last day of the month.

Agent

Advance NDC can be configured to run the automatic INIT when called by third-party software or batch file. This is configured using the Configure menu in Supervisor mode.

This sets the following registry key:

```
HKLM\SOFTWARE\NCR\Advance NDC\EJOptions\EJAgent
```

Valid values are as follows:

- 0 for disabled. This is the default.
- 1 for enabled.

To run successfully:

- The third-party software or batch file must be installed in the appropriate directory. For SST installations, this is *C:\ssds*. For development installations, the default is *<drive>:\ntglobal.*
- The agent automatic INIT must be enabled
- Advance NDC must be running.

Once these pre-requisites are met, the agent can run an automatic INIT as described in the following sections.

Third-Party Software Third party software must call *InitEJ.dll*:

```
extern 'C' IMP_OR_EXP BOOL EJAgentInit()
```

which is declared in *InitEH.h*.

Batch File If using a batch file, you have to use *InitEJ.exe* to call *InitEJ.dll.*

The batch file can be called by remote third-party software or run from the command line at the SST. Any error codes are returned to the batch file.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

The following is an example of a batch file used to perform an automatic INIT:

Figure 5-2
Example Batch File

```
@echo off

::InitEJ.exe

InitEJ.exe
if errorlevel 1 goto errhand
echo EJ Initialised
goto end

:errhand
echo EJ failed to initialise
:end
```

Automatic INIT Copy Drive

Advance NDC can be configured to make a copy of the automatic INIT EJ file using the Configure menu in Supervisor mode.

This sets the following registry key:

```
HKLM\SOFTWARE\NCR\Advance NDC\EJOptions\SecondaryBkp
```

This key stores the drive letter of the selected destination.

The destination drive must be writable, however, it cannot be the diskette drive.

EJ Compression

Advance NDC can be configured to compress the EJ file on performing an INIT using the Configure menu in Supervisor.

This sets the following registry key:

```
HKLM\SOFTWARE\NCR\Advance NDC\EJOptions\EJCompression
```

Valid values are as follows:

- 0 for disabled. This is the default.
- 1 for enabled.

EJ Privacy

If the password is forgotten, the EJ files can no longer be accessed. To allow access to future EJ files, you must deinstall Advance NDC, delete the *UCDIPers.dat* file, and reinstall Advance NDC.

The *UCDIPers.dat* file is located in the *C:\Program Files\NCR APTRA\Advance NDC\Data* directory.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Maximum EJ File Size

The EJ file size can be configured using the Configure menu in Supervisor.

This sets the following registry key:

```
HKLM\SOFTWARE\NCR\Advance NDC\EJOptions\EJFileSize
```

In Supervisor, the value is entered in kilobytes. The default is 1411000 bytes, or 1378 KB, for backward compatibility. The minimum value is 1 KB. The maximum value is calculated in the following way:

```
available disk space / (maximum number of backups + 4)
```

Disabling Host Control of Enhanced EJ Backup

The EJ backup mode can be changed either through option 36 of the Enhanced Configuration Parameters Download, or through Supervisor. Sometimes local control is required and the host control can be disabled using the Configure menu in Supervisor mode.

This sets the following registry key:

```
HKLM\SOFTWARE\NCR\Advance NDC\EJOptions\EJBackupECP
```

Valid values are as follows:

- 0 to prevent the host from changing the EJ backup mode
- 1 to allow the host to change the EJ backup mode. This is the default.

EJ Checksum

A checksum is added to the EJ file by default. This cannot be changed through Supervisor. However, if required, the checksum can be disabled by adding the following registry key:

```
HKLM\SOFTWARE\NCR\Advance NDC\EJOptions\EJChecksum
```

Valid values are as follows:

- 0 to disable the checksum
- 1 to enable the checksum

As this registry key would only be added to disable the checksum, there is no default value for the key. The checksum is added when no registry value is present. Removing the registry key reinstates the checksum.

Journal Level

The information that is journalled for a transaction can be configured using the Configure menu in Supervisor.

The following registry key (`DWORD`) defines what is journalled:

```
HKEY_LOCAL_MACHINE\SOFTWARE\NCR\Advance NDC\JournalLevel
```

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Valid values for this key are 0, 1, 2 and 3. For a full description of the values, refer to the *APTRA Advance NDC, Supervisor's Guide*.

BNA Encash, Print and Set Next State

The transaction reply printer data can be sent to the printer at the same time as the encash operation is carried out. This results in a parallel print and encash.

To enable parallel print and encash, the following registry key must be set to 1:

```
HKLM\SOFTWARE\NCR\Advance NDC\BNA\DoParallelEncashPrint
```

Valid values for this key are as follows:

- 0 to disable parallel print and encash. This is the default
- 1 to enable parallel print and encash.

BNA Count Journal Format

The format for the journalling of BNA counts can now be configured to display in an alternate format described in the *APTRA Advance NDC, Supervisor's Guide*.

Advance NDC provides the registry keys described in to enable this alternate format.

Table 5-15
BNA Count Journal Format Registry Keys

Registry Key	Description	Value
HKLM\SOFTWARE\NCR\Advance NDC\BNA\CashInJPTR\ESCROW_H	Defines the display identifying escrow counts	ESC:
HKLM\SOFTWARE\NCR\Advance NDC\BNA\CashInJPTR\Report0Counters	Enables or disables the reporting of all enabled note types	1
HKLM\SOFTWARE\NCR\Advance NDC\BNA\CashInJPTR\ReportCustomerData	Enables or disables the reporting of customer information	0
HKLM\SOFTWARE\NCR\Advance NDC\BNA\CashInJPTR\VAULT_H	Defines the display identifying vaulted counts	VAL:
HKLM\SOFTWARE\NCR\Advance NDC\BNA\CashInJPTR\RETURN_H	Defines the display identifying returned counts	RET:
HKLM\SOFTWARE\NCR\Advance NDC\BNA\CashInJPTR\TCODE0	Defines the error message for a TCode of 0	OPERATION OK
HKLM\SOFTWARE\NCR\Advance NDC\BNA\CashInJPTR\TCODE1	Defines the error message for a TCode of 1	CUSTOMER CANCEL TRANSACTION
HKLM\SOFTWARE\NCR\Advance NDC\BNA\CashInJPTR\TCODE2	Defines the error message for a TCode of 2	CASH BIN FULL

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Registry Key	Description	Value
HKLM\SOFTWARE\NCR\Advance NDC\BNA\CashInJPTR\TCODE3	Defines the error message for a TCode of 3	CASHUNIT FAILED
HKLM\SOFTWARE\NCR\Advance NDC\BNA\CashInJPTR\TCODE4	Defines the error message for a TCode of 4	BILLS AT EXIT
HKLM\SOFTWARE\NCR\Advance NDC\BNA\CashInJPTR\TCODE5	Defines the error message for a TCode of 5	ESCROW NO BILL
HKLM\SOFTWARE\NCR\Advance NDC\BNA\CashInJPTR\TCODE6	Defines the error message for a TCode of 6	BILL AT POWERUP
HKLM\SOFTWARE\NCR\Advance NDC\BNA\CashInJPTR\TCODE7	Defines the error message for a TCode of 7	BILLS RETRACTED
HKLM\SOFTWARE\NCR\Advance NDC\BNA\CashInJPTR\TCODE8	Defines the error message for a TCode of 8	ERROR NO ACCESS
HKLM\SOFTWARE\NCR\Advance NDC\BNA\CashInJPTR\TCODE9	Defines the error message for a TCode of 9	UNABLE TO ENCASH
HKLM\SOFTWARE\NCR\Advance NDC\BNA\CashInJPTR\TCODEA	Defines the error message for a TCode of A	UNABLE TO REFUND
HKLM\SOFTWARE\NCR\Advance NDC\BNA\CashInJPTR\TCODEB	Defines the error message for a TCode of B	FULL -ENCASHFAIL
HKLM\SOFTWARE\NCR\Advance NDC\BNA\CashInJPTR\TCODEC	Defines the error message for a TCode of C	FULL - CASHJAMMED
HKLM\SOFTWARE\NCR\Advance NDC\BNA\CashInJPTR\NumberOfColumns	Defines the width of the journal paper	16
HKLM\SOFTWARE\NCR\LOCAL CONFIGURATION\CashInJPTRDenominationNames	Allows the denomination name printed on the journal to be changed from the default	Blank
HKLM\SOFTWARE\NCR\Advance NDC\BNA\CashInJPTR\ReportCustomerData *See* Table Note 5	Defines the journalling of the cardholder PAN. Valid values are as follows: ● 0 to omit the cardholder PAN from the journal ● 1 to journal the cardholder PAN when a device error is sent (a TCode of 3) ● 2 to journal the cardholder PAN when a device status is sent	0

Table Note 5: To journal the cardholder PAN, the PANDX field must be configured in the FIT as described in the *APTRA Advance NDC, Reference Manual*.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Journalling Retract Counts

Retract counts are journalled after a retract operation, as described in the *APTRA Advance NDC, Supervisor's Guide.*

The journalled information can be edited using the *CashInJournalFormatting.wsc* file; for example, to change the format or for localisation.

To display the list of note types, by default listed under RECOGNIZED ITEMS, the SP must be configured to report the note list on the retract bin by setting the following registry key:

```
HKLM\SOFTWARE\NCR\XFS GBRU-GBNA Service Provider\
XFS-DeviceControl\GBRU-GBNA\Interoperability\
ExpandCIMRetractNoteList
```

To ensure that the retract serial number is correctly updated, the SP must be configured to report retract operations by setting the following registry key:

list on the retract bin by setting the following registry key:

```
HKLM\SOFTWARE\NCR\XFS GBRU-GBNA Service Provider\
XFS-DeviceControl\GBRU-GBNA\Interoperability\
ReportRetractOperations
```

Journalling Reject Counts

Reject counts are journalled as part of the returned counts journalling, as described in the *APTRA Advance NDC, Supervisor's Guide.*

You can control the "REJECTS" label, for example for localisation, using the following registry key:

```
HKLM\SOFTWARE\NCR\Advance NDC\BNA\CashInJPTR
```

By default, this is set to REJECTS.

Cash In Component

This section includes configuration information for the Cash In Component.

Registry Settings that Must Not Be Changed

The following registry settings are set during installation and must not be changed:

```
HKLM\SOFTWARE\NCR\XFS CIM Service Provider\
XFS Device Control\CIM\Interoperability\ReportSeparateCashUnits

HKLM\SOFTWARE\NCR\Advance NDC\BNA\NativeMstatus
```

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Enhanced Notes Accepted Screen

By default, the notes accepted screen lists 50 note types with the number of notes entered for each using screens M10 and M11. It is displayed after the cardholder has entered notes to confirm the amount deposited.

Advance NDC can be configured to display screens U0104 and U0105 that list only the deposited notes with user-friendly descriptions of the currency, value and number of notes deposited. Up to eight lines of note information can be displayed.

The screen to use is controlled using the following registry key:

```
HKLM\SOFTWARE\NCR\Advance NDC\BNA\EnhancedConfirmationScreen
```

Valid values are as follows:

- 0 to list all 50 notes types. This is the default and provides compatibility with NDC+
- 1 to list the deposited notes with currency, value, number of notes for each note type and total number of notes for each currency.

For further information on the Confirmation screen, refer to the *APTRA Advance NDC, Reference Manual.*

You can control the “TOTAL” label, for example for localisation, using the following registry key:

```
HKLM\SOFTWARE\NCR\Advance
NDC\BNA\ConfirmationScreenStringforTotal
```

By default, this is set to `TOTAL`.

Whether to display delimiters in amount entry is set using the following registry key:

```
HKLM\SOFTWARE\NCR\Advance NDC\BNA\ConfirmationScreenGroupDigit
```

Valid values are as follows:

- 0 to display the amount without a delimiter, for example 4500. This is the default
- 1 to display the amount with a delimiter, for example 4,500.

GBXX Dynamic Note Sorting

Advance NDC can be configured to allow dynamic note sorting in GBXX devices if a cassette becomes full. This effectively increases the capacity of the device, as notes can continue to be accepted.

If an ALLIN cassette is configured, it is used when any other cassette becomes full and no reconfiguration will occur until this cassette also becomes full.

Advance NDC reconfigures the priority cassette in the Close state after a high cassette event. Following this reconfiguration, once the

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

cassette becomes full, notes are automatically sorted to the newly reconfigured cassette. The priority sequence is defined by the following registry key:

```
HKLM\SOFTWARE\NCR\Advance NDC\BNA\
CashinCassetteReconfigurePriority
```

This key must be created and should contain a REG_SZ value. Valid values for this key are the cassette identifiers listed in priority order.

Note: The registry value must not contain any spaces.

The following scenarios illustrate how the dynamic note sorting occurs.

Note: As the reconfiguration occurs during the Close state, there are some scenarios where the reconfiguration cannot take place. For example, if a cassette becomes full during a chained transaction.

In all of these scenarios, the desired priority is Cassette 1, Cassette 2, Cassette 3, and then Cassette 4, and the registry value is as follows:

```
CI1,CI2,CI3,CI4
```

Scenario 1: No ALLIN Cassette Configured

The cassettes are configured as follows:

- Cassette 1 (CI1) accepts 10 Euro notes
- Cassette 2 (CI2) accepts 20 Euro notes
- Cassette 3 (CI3) accepts 50 Euro notes
- Cassette 4 (CI4) accepts 100 Euro notes

If Cassette 1 becomes full, Cassette 2 will be dynamically configured to accept 10 Euro notes. Cassette 2 is reconfigured as it is the next cassette in the priority list.

If Cassette 1 is not full but Cassette 2 becomes full, Cassette 1 will be dynamically configured to accept 20 Euro notes. In this case, Cassette 1 is reconfigured as it has the highest priority.

Scenario 2: ALLIN Cassette Configured

All cassettes are configured as follows:

- Cassette 1 (CI1) accepts 10 Euro notes
- Cassette 2 (CI2) accepts 20 Euro notes
- Cassette 3 (CI3) accepts 50 Euro notes
- Cassette 4 (CI4) accepts all notes (ALLIN cassette)

If Cassette 1 becomes full, notes will be sorted to Cassette 4 with no reconfiguration.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

If Cassette 4 also becomes full, Cassette 2 will be dynamically configured to accept 10 Euro notes.

GBXX Note Reporting

Denominations are reported in the same order until the order in the *GBNA.ini* file is changed. The *GBNA.ini* file can be used to order and select or deselect a subset of the available denominations.

If this file is changed, Central should request the note definitions again.

The default location for the *GBNA.INI* file is *C:\Program Files\NCR\APTRA\Advance NDC\data*, however, this can be changed using the following registry key:

```
HKLM\Software\NCR\Advance NDC\BNA\GBNAPath
```

In the following example *GBNA.ini* file:

- The first two digits are the country code
- The next three digits are the note ID
- EURxxx is the description of the denominations
- A hyphen (-) shows that the denomination is accepted by the device and included in the report to the host (active)
- A number sign (#) shows that the denomination is rejected by the device and not included in the report to host (deactive).

Note: Whether to include a specific denomination in the report to the host is set using the Supervisor Config menu. For more details, refer to the *APTRA Advance NDC Supervisor's Guide.*

Note: Denomination one is the denomination in position one in the array of note types. In the Cash Accept State, the first active denomination in the *GBNA.ini* file is used as denomination one. For example, in Figure 5-3, the first active denomination is 50 Euros, therefore 50 Euros is used as denomination one. For more details, refer to the *APTRA Advance NDC, Reference Manual.*

Figure 5-3
Example *GBNA.ini* File

```
23,259,EUR5#
23,260,EUR10#
23,261,EUR20#
23,262,EUR50-
23,263,EUR100#
23,264,EUR200-
23,265,EUR500-
```

GBXX Cassette Configuration

To configure the cassettes for a GBXX device, you must use an XML configuration file and associated XML schema.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Note: The configuration file can be used only with GBXX devices, and is not applicable for BNA devices.

The configuration file is called using 39 - GBRU / GBNA CONFIGURATION on the Configure menu in Supervisor. For details, refer to the *APTRA Advance NDC, Supervisor's Guide*. This option will fail if the configuration file does not exist, or does not conform to the guidelines provided here.

An example configuration file is given in "GBXX Configuration File" on page 5-54. The file can be created and edited in any text editor, but no template or sample file is supplied with Advance NDC. Once created, the configuration file must be installed to `C:\Program Files\NCR APTRA\Advance NDC\data` using your preferred method.

The configuration file is an XML file, which must conform to and reference the XML schema given in "XML Schema" on page 5-48.

You must create the configuration file, and include the following information:

- Currency
- Denomination
- Initial counts

  Note: The initial counts in the NoteCount element must be set to 0, the counts are updated by the ADD CASH replenishment option in Supervisor mode.

- Number of notes to accept before reporting the high bin threshold
- List of notes that can be deposited in particular cassettes
- Settings for the `GBRUMode` registry key
- Settings for the `ValidationMode` registry key.

XML Schema

The following schema must be referenced, as shown in "GBXX Configuration File" on page 5-54:

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:xs="http://www.w3org/2001/XMLSchema">
 <xs:complexType name="NoteIDSType">
  <xs:sequence>
   <xs:element name="NoteID" type="xs:string"
minOccurs="0"/>
  </xs:sequence>
 </xs:complexType>
 <xs:complexType name="CassetteType">
   <xs:sequence>
    <xs:element name="CassetteID" type="xs:string"
minOccurs="1" maxOccurs="1"/>
    <xs:element name="CurrencyType" type="xs:string
```

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

```
minOccurs="1" maxOccurs="1""/>
    <xs:element name="CurrencyValue" type="xs:string"
minOccurs="1" maxOccurs="1"/>
    <xs:element name="NoteCount" type="xs:string"
minOccurs="1" maxOccurs="1"/>
    <xs:element name="NoteIDS" type="NoteIDSType"
minOccurs="1" maxOccurs="1"/>
    <xs:element name="MaxCashInItems" type="xs:string"
minOccurs="0" maxOccurs="1"/>
   </xs:sequence>
 </xs:complexType>
 <xs:complexType name="ConfigDetailsType">
   <xs:sequence>
    <xs:element name="Cassette" type="CassetteType"
maxOccurs="unbounded"/>
   </xs:sequence>
   <xs:attribute name="ConfigID" type="xs:string"
use="required"/>
 </xs:complexType>
 <xs:complexType name="ActiveConfigType">
   <xs:sequence>
    <xs:element name="ActiveConfigID"
type="xs:string"/>
    <xs:element name="GBRUMode" type="xs:string"/>
    <xs:element name="ValidationMode type="xs:string"/>
   </xs:sequence>
 </xs:complexType>
 <xs:element name="GBRUConfig">
  <xs:complexType>
   <xs:sequence>
    <xs:element name="ActiveConfig"
type="ActiveConfigType"/>
    <xs:element name="ConfigDetails"
type="ConfigDetailsType" maxOccurs="unbounded"/>
   </xs:sequence>
  </xs:complexType>
 </xs:element>
</xs:schema>
```

Schema Elements The schema elements are described in the following table.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Table 5-16
XML Schema Elements

Element Name	Description
GBRUConfig	The whole file is embedded within `GBRUConfig` tags. This is the root element. Nested elements: ● ActiveConfig ● ConfigDetails
ActiveConfig	Mandatory. Appears once only. Defines the mode of operation and validation for a GBXX device. Defines the configuration to use for the current GBXX device. Nested elements: ● ActiveConfigID ● GBRUMode ● ValidationMode
ActiveConfigID	Mandatory. Identifies the configuration set to use when configuring the current GBXX device.
GBRUMode	Mandatory for GBRU. Ignored for GBNA. Identifies the mode in which a GBRU device works. Valid values are as follows: 0 - Recycling mode. Cash can be deposited to and dispensed from a recycling cassette. Advance NDC does not support recycling. 1 - Cash Dispense mode. Cash can only be dispensed from cash-out cassette 2 - Cash Deposit mode. Cash can only be deposited to cash-in cassettes 3 - Non-recycling mode. Cash can be deposited to a cash-in cassette, and dispensed from a cash-out cassette, but these must be separate cassettes **Note:** If using the GBRU as a Cash-In / Cash-Out device, this must be set to 3

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Element Name	Description
ValidationMode	Mandatory. Identifies the currency validation mode of the GBXX device. Valid values are as follows: 0 - Actual notes 1 - Test notes
ConfigDetails	Mandatory. Appears one or more times. Defines a complete set of configuration parameters to use when configuring a GBXX device. Each `ConfigDetails` sub-element is identified by a `ConfigID` attribute. This `ConfigID` attribute is used by the `ActiveConfig` sub-element to determine the configuration to use. Nested elements: ● Cassette ● ConfigID
ConfigID	Mandatory. Identifies a configuration set to be used for configuring the GBXX device.
Cassette	Mandatory. Defines the configuration parameters for each cassette that is to be configured. Nested elements: ● CassetteID ● CurrencyType ● CurrencyValue ● NoteCount ● MaxCashInItems ● NoteIDS

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Element Name	Description
CassetteID	Mandatory. Identifies the cassette; for example CI15, RC10, or BC5. All cassette types present in the GBXX must be defined in the configuration file with a `CassetteID` entry. If an entry is not found for an existing cassette type, an error is produced when running the Supervisor option. Entries can be included for cassette types not currently present in the SST to avoid reworking the XML file in the future. **Note:** If a recycling cassette is identified and the mode is changed such that recycling is not supported, the `CassetteID` should be updated to identify a bill cassette. For example, RC1 would be updated to BC1. All other configuration settings can remain the same.
CurrencyType	Mandatory. Identifies the currency to be accepted or dispensed by the cassette. If `CurrencyType` is empty and `NoteID` is specified, the cassette is configured to accept the specified note. If `CurrencyType` is empty and `NoteID` is empty, the cassette is configured to accept all notes listed within the NoteIDS element. `CurrencyType` uses the ISO-4217 format. Settings must not conflict with the Advance NDC cash dispenser mapping registry settings.
CurrencyValue	Mandatory. Identifies the value of the currency to be accepted or dispensed by the cassette. If `CurrencyValue` is empty and `CassetteID` points to a cash-in cassette, the cassette is configured to accept all valid notes specified by `CurrencyType`. `CurrencyValue` should be left empty to configure a cash-in cassette that accepts all notes listed within the NoteIDS element.
NoteCount	Mandatory. Specifies the number of notes available in the cassette. This value must initially be set to zero (0). Running ADD CASH in Supervisor mode populates the counts.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Element Name	Description
NoteIDS	Mandatory. Defines one or more note identifiers for the currency and value specified in `CurrencyType` and `CurrencyValue`. If multiple note identifiers are specified, the CurrencyType and CurrencyValue are ignored. Nested elements: ● NoteID
NoteID	Optional. Identifies a note ID for the currency and value specified in `CurrencyType` and `CurrencyValue`. Note ID values are vendor-specific.
MaxCashInItems	Optional. Specifies the maximum number of notes that can be accepted before the SP reports a cassette high threshold. If the `MaxCashInItems` element is present but empty, the current setting is retained. If the cassette is a bill cassette (type BC or CI), and the `MaxCashInItems` element is not present, the default of 2000 is used If the cassette is a retract, reject, or counterfeit bin, the following apply: ● When running the GBXX configuration file, only the `MaxCashInItems` element is used ● If the `MaxCashInItems` element is not present, the default of 200 is used

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

GBXX Configuration File

The following is an example GBXX configuration file:

```
<?xml version="1.0" encoding="utf-8"?>
<GBRUConfig
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="GBRUConfig.xsd">
 <ActiveConfig>
  <ActiveConfigID>2</ActiveConfigID>
  <GBRUMode>0</GBRUMode>
  <ValidationMode>1</ValidationMode>
 </ActiveConfig>
 <ConfigDetails ConfigID="1">
 <Cassette>
  <CassetteID>CI15</CassetteID>
  <CurrencyType/>
  <CurrencyValue/>
  <NoteCount>0</NoteCount>
  <NoteIDS/>
  <MaxCashInItems>2000</MaxCashInItems>
 </Cassette>
 <Cassette>
  <CassetteID>BC1</CassetteID>
  <CurrencyType>EUR</CassetteType>
  <CurrencyValue>10</CurrencyValue>
  <NoteCount>0</NoteCount>
  <NoteIDS>
   <NoteID>260</NoteID>
  </NoteIDS>
  <MaxCashInItems>2000</MaxCashInItems>
 </Cassette>
 <Cassette>
  <CassetteID>BC2</CassetteID>
  <CurrencyType>EUR</CurrencyType>
  <CurrencyValue>20</CurrencyValue>
  <NoteCount>0</NoteCount>
  <NoteIDS>
   <NoteID>261</NoteID>
  </NoteIDS>
  <MaxCashInItems>2000</MaxCashInItems>
 </Cassette>
 <Cassette>
  <CassetteID>BC3</CassetteID>
  <CurrencyType>EUR</CurrencyType>
  <CurrencyValue>50</CurrencyValue>
  <NoteCount>0</NoteCount>
  <NoteIDS>
   <NoteID>262</NoteID>
  </NoteIDS>
  <MaxCashInItems>2000</MaxCashInItems>
 </Cassette>
</ConfigDetails>
<ConfigDetails ConfigID="2">
 <Cassette>
  <CassetteID>CI15</CassetteID>
  <CurrencyType/>
  <CurrencyValue/>
  <NoteCount>0</NoteCount>
  <NoteIDS/>
  <MaxCashInItems>2000</MaxCashInItems>
```

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

```
 </Cassette>
 <Cassette>
  <CassetteID>RC1</CassetteID>
  <CurrencyType>EUR</CassetteType>
  <CurrencyValue>10</CurrencyValue>
  <NoteCount>0</NoteCount>
  <NoteIDS>
   <NoteID>260</NoteID>
  </NoteIDS>
  </Cassette>
 <Cassette>
  <CassetteID>RJ1</CassetteID>
  <CurrencyType></CurrencyType>
  <CurrencyValue></CurrencyValue>
  <Notecount></NoteCount>
  <NoteIDS></NoteIDS>
  <MaxCashInItems></MaxCashInItems>
 </Cassette>
 <Cassette>
  <CassetteID>RT1</CassetteID>
  <CurrencyType></CurrencyType>
  <CurrencyValue></CurrencyValue>
  <NoteCount></NoteCount>
  <NoteIDS>
  </NoteIDS>
 </Cassette>
</ConfigDetails>
</GBRUConfig>
```

GBRU Registry Settings

The following registry configuration is needed to support GBRU cash-in and cash-out functionality.

Table 5-17 shows the keys set by Advance NDC, and Table 5-18 shows the keys that you must update to complete the configuration.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Table 5-17
GBRU Registry Keys Set by Advance NDC

Key	Description	Value
HKLM\Software\NCR\XFS GBRU-GBNA Service Provider\XFS-DeviceControl\GBRU-GBNA\Operations\GBRUMode	Set at installation to set the GBRU mode. For details of the GBRU modes see Table 5-16 on page 5-50.	3
HKLM\Software\NCR\XFS GBRU-GBNA Service Provider\XFS-DeviceControl\GBRU-GBNA\Operations\SuspendTimeout	Set at installation to set the GBRU suspend timeout. Advance NDC uses the maximum value to ensure that it has control over the GBRU timeouts.	900
HKLM\Software\NCR\XFS GBRU-GBNA Service Provider\XFS-DeviceControl\GBRU-GBNA\Interoperability \AllowResetDuringSuspend	Set at installation to allow a reset command to be sent when the GBRU is in suspend state.	1
HKLM\Software\NCR\XFS GBRU-GBNA Service Provider\XFS-DeviceControl\GBRU-GBNA\Interoperability\SuppressManipStatus	Set at installation to ensure that the MANIP state is not reported when a cassette is removed.	1
HKLM\Software\NCR\XFS GBRU-GBNA Service Provider\XFS-DeviceControl\GBRU-GBNA\ Interoperability\NotRejectOnDispense	Set at installation to prevent notes found in the stacker when WFS_CMD_CDM_DISPENSE is called from being automatically sent to the reject area.	1
HKLM\Software\NCR\XFS GBRU-GBNA Service Provider\XFS-DeviceControl\GBRU-GBNA\ Interoperability\RejectOnCashInStart	Set at installation to prevent notes found in the stacker when WFS_CMD_CIM_CASH_IN_START is called from being automatically sent to the reject area.	0
HKLM\Software\NCR\APTRA Self-Service Support (NCR Features)\GBRU\Operational Parameters\ DisableAutoResetErrorRecovery	Set at installation to prevent automatic error recovery when a hardware error occurs.	1
HKLM\Software\NCR\APTRA Self-Service Support (NCR Features)\GBRU\Operational Parameters\ClearTransportOnStartUp	Set at installation to ensure that notes are not cleared from the transport at startup.	0
HKLM\Software\NCR\Advance NDC\CashRecycler\BillDispenser	Set at runtime if the device is a GBRU.	1
HKLM\Software\NCR\Advance NDC\ CashRecycler\CashRecyclerMapping	Set at runtime depending on the setting of Enhanced Configuration option 78.	
	If option 78 is 000	0
	If option 78 is 001	1

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Key	Description	Value
HKLM\Software\NCR\Advance NDC\CashRecycler\ResetRequired	Set at runtime to identify whether an error has occurred, and whether any error was on a cash-in or a cash-out transaction.	
	If there was no error	0
	If there was an error during a cash-out transaction	1
	If there was an error during a cash-in transaction	2
HKLM\Software\NCR\Advance NDC\CashRecycler\TransactionType	Set at runtime to identify whether the last transaction was a cash-in or a cash-out transaction.	
	If the last transaction was a cash-out transaction	0
	If the last transaction was a cash-in transaction	1

The following keys must be updated to use the stated values to complete the configuration.

Table 5-18
GBRU Registry Keys to Update

Key	Description	Value
HKLM\Software\NCR\Advance NDC\Aliases	Specifies that the cash handler is a GBRU	BillDispenser1
HKLM\Software\NCR\Advance NDC\PhysicalCassettes\TOP	Maps the physical position name used in a GBRU to those used in a cash dispenser	Cassette 1
HKLM\Software\NCR\Advance NDC\PhysicalCassettes\SECOND	Maps the physical position name used in a GBRU to those used in a cash dispenser	Cassette 2
HKLM\Software\NCR\Advance NDC\PhysicalCassettes\THIRD	Maps the physical position name used in a GBRU to those used in a cash dispenser	Cassette 3

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Key	Description	Value
HKLM\Software\NCR\Advance NDC\PhysicalCassettes\BOTTOM	Maps the physical position name used in a GBRU to those used in a cash dispenser	`Cassette 4`
HKLM\Software\NCR\Advance NDC\Supervisor	Ensures that the GBRU is cleared using a reset rather than a reject command after the following: ● suspend ● exit from Supervisor ● Test Cash operation ● STD Cash operation	`0x141`
HKLM\Software\NCR\Advance NDC\CurrencyTable	Defines the currency types.	Must match the contents of the *GBRUConfig.xml* file.

Cassettes can be either cash-in (for cash accept transactions) or recycling (for dispense transactions). To use a GBRU for Cash-In / Cash-out, at least one cassette should be a cash-in, and at least one a recycling cassette. The reject bin contains three partitions, as follows:

- Reject area
- Retract area
- Counterfeit area.

Maximum Accepted Notes

If you use Enhanced Configuration option 45 to limit the number of accepted notes to 90 and also use any GBRU devices, you must reduce the number of notes accepted by the device to 90.

To limit the number of accepted notes, update the usmaxAcceptednote parameter in the *GBRU_DP2_initial.ini* file. This parameter is set by default to 200.

The file is located in *C:\Program Files\NCR APTRA\PcGBRU* and can be edited using a text editor.

Enabling Web Exits

Advance NDC can use a web exit to display screens and interact with the consumer outside of Advance NDC. The Advance NDC state flow is exited and full control passed to a web browser with no need for a host change to the existing state flow.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Enabling web exits involves the following:

1 Altering the state flow to introduce the new state.

2 Activating the web exit state.

Exiting the web exit state and passing control back to Advance NDC.

Configuring the Web State

The web state is configured using the following steps:

1 Assign any available state type letter. This is used to represent the state.

2 Associate state type parameters containing the state type and state information for each web exit state that can be executed.

3 Populate the state table parameters with the updated web state parameters using local customisation.

Appendix G, “Web Exit State” provides the following information on enabling web exits:

State table parameters. State table parameters control the behaviour of the web exit.

Tracking URLs. Tracking URLs communicate the state of the web browsing to Advance NDC.

The URL index format. The URL index defines the locations of the local customisation screens to be used in the web exit. For details of providing local customisation data see “Specifying Local Customisation Data” on page 5-59.

Updating STCONT. STCONT registers the state type letter and location of the implementation of the web exit.

Updating the state flow. The state flow allows the exit to be called at the appropriate time.

Specifying Local Customisation Data

Local customisation data is used during the web exits. It can also be specified independently to override information in the host download with locally provided data

Local customisation data allows files placed directly on the SST to override or modify the download sent by the host.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

To use local customisation data, the following registry key must be set to '1':

```
HKLM\Software\NCR\Advance NDC\Local Data Available
```

The files containing the customisation data are either defined within the *SCXLOC* file or contained within the *LOCAL* file.

The *SCXLOC* file takes precedence, so if it is present on the SST, any *LOCAL* file is ignored.

If neither file is found, no local customisation can take place.

SCXLOC File

SCXLOC is a control file located on the SST in the *C:\SSDS\DLL* directory. The file contains a list of files to be applied as local customisation files, allowing you to define local customisation groups as separate files.

Each file is listed on a separate line, and the maximum file size for SCXLOC is 1024 bytes.

All the files referenced must also be placed in the *C:\SSDS\DLL* directory on the SST. As some methods of file copying do not support filenames with spaces you should avoid using spaces in the filenames.

Any screens, states or FITs that are part of the host download, but are not being locally customised should not be included in these files.

The referenced files can contain multiple customisations, but each customisation can be no more than 1024 bytes.

Example SCXLOC File The following is an example of an SCXLOC file, referencing 4 files:

```
LOCSC1
LOCSC2
LOCST1
LOCFI1
```

Example Referenced Files The following list provides examples of the four files referenced in the example SCXLOC file.

The tags are defined in Table 5-19.

Table 5-19
Tags Used in Files

Tag	Hex Code	Description
<fs>	1CH	Field separator

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Tag	Hex Code	Description
<ff>	0CH	Clear screen character
<si>	0FH	Shift in cursor position character
<ex>	03H	End of text character

- LOCSC1:

  The following example defines the locally customised screen identified as 010:

  ```
  3<fs><fs><fs>11<fs>010<ff><si>DDPLEASE INSERT YOUR
  CARD<si>FDTO BEGIN A TRANSACTION<ex>
  ```

- LOCSC2:

  The following example defines the locally customised screen identified as 11:

  ```
  3<fs><fs><fs>11<ff><si>EGWE ARE UNABLE<si>GDTO
  PROCESS YOUR CARD<ex>
  ```

- LOCST1:

  The following example defines the locally customised states identified as 000 and 095:

  ```
  3<fs><fs><fs>12<fs>000A0100010110020020020010 95<fs>095J0110000110
  11011000011000<ex>
  ```

- LOCFI1:

  The following example defines the locally customised FIT table identified as 000:

  ```
  3<fs><fs><fs>15<fs>000000255255255255255000000012000015000000
  000255000000000000000000000000000000000000000000000000000000
  00000<ex>
  ```

LOCAL File

The LOCAL file contains all locally customised screens, states and FIT Tables in a single file. The file is located in *C:\SSDS\DLL*.

Any screens, states or FITs that are part of the host download, but are not being locally customised should not be included in this file.

The file is in message file format, which mimics the download from the host.

The maximum length of each screen, state or FIT is 1024 bytes.

Example LOCAL File The following is an example of a LOCAL file, with definitions for locally customised screens identified as 010

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

and 011, states identified as 000 and 095, and a FIT table identified as 000.

For an explanation of the tags used, see Table 5-19.

```
3<fs><fs><fs>11<fs>010<ff><si>DDPLEASE INSERT YOUR
CARD<si>FDTO BEGIN A TRANSACTION<fs>
011<ff><si>EGWE ARE UNABLE<si>GDTO PROCESS YOUR
CARD<ex>3<fs><fs><fs>12<fs>000A0100010110020020020010 95<fs>095J
0110000110110110000110 00<ex>3<fs><fs><fs>15<fs>000000255255255255
2550000000120000150000000002550000000000000000000000000000000000
0000000000000000000000000<ex>
```

Additional DASH Reader Fatal/Suspend Handling

When using a DASH reader with Card Read State A, or Card Read State T with no chip connect bit set, the reader acts as a standard DIP reader and does not enter a suspend state.

To use the additional fatal/suspend handling the following registry key must be set to '1':

```
HKLM\Software\NCR\Advance NDC\DASH\EnableSeverity
```

By default, this option is not enabled.

When enabled, the following additional fatal/suspend handling occurs:

- If a card is entered in a DASH reader and not removed before Timer 72 expires, a suspend message is sent and the DASH reader enters a suspend timeout.
- If the card is still detected in the DASH reader after the suspend timeout expires, the reader reports a fatal status and goes out of service. The card entry screen is displayed, but no cards can be accepted.

Note: Because the suspend and fatal states are initiated by the application, they are not reflected in the fault display information.

Barcode Filter Configuration

The barcode reader configuration is defined within the *NDCBarcodeReader.xml* file. This xml-based configuration file can be updated through software distribution, local modification or the XML Configuration download. For details of the XML configuration download refer to the *APTRA Advance NDC, Reference Manual*.

The *NDCBarcodeReader.xml* file is stored in the following location:

C:\Program Files\NCR APTRA\Advance NDC\Config

The Barcode Filter schema, *NDCBarcodeReader.xsd*, defines the structure of the configuration data for the barcode reader.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and/or distribution is strictly prohibited.

Table 5-20
Barcode Filter Structure Definition

Element	Description
BarcodeReaderClass	The whole file is embedded within `BarcodeReaderClass` tags. This is the root element. Nested elements: ● `BarcodeReaderDevice`
BarcodeReaderDevice	The identification number of the barcode reader device. Nested elements: ● `Name` ● `Instance` ● `BarcodeFilter`
Name	A free format name assigned to the device. This name is for information only.
Instance	The barcode reader device that the configuration applies to. There is only one barcode reader available for the barcode reader device class.
BarcodeFilter	A list of barcode filters that are used for comparison within the Barcode Read state. These filters ensure that the barcode conforms to an application template and allow the state flow to branch, based on the barcode presented. Nested elements: ● `Index` ● `Offset` ● `Filter`
Index	This value is used as an indice for the Barcode Read state parameters and is used to select an exit. `Index` is in the range 0 to 48. *See* section "&-Barcode Reader State" in the *APTRA Advance NDC, Reference Manual.* **Note:** The values should be kept as low as possible to minimize the number of extension states required. The most common filters are assigned lower range index identifier values.
Offset	The character position offset within the barcode where the comparison starts.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Element	Description
Filter	Specifies a character filter, which ensures that only barcodes of the correct format are accepted by the application. `Filter` can contain any ASCII hex character from 0x20 to 0x7E. The following characters have a special meaning within the filter process: ● @ = numeric character 0-9 ● % = any character from 0x20 to 0x7E For example, 'abc@@@@%%%def' would be used to select a barcode which is a string containing the following sequence: **1** The characters 'abc' **2** Any four numeric characters in the range 0-9 **3** Any three characters from 0x20 to 0x7E **4** The characters 'def'.

Example Barcode Filter Configuration File

The following is an example Barcode Filter configuration file:

```
<?xml version="1.0" encoding="UTF-8"?>
<BarcodeReaderClass
xmlns="http://www.ncr.com/APTRA/NDC/BarcodeReader"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.ncr.com/APTRA/NDC/BarcodeReader
NDCBarcodeReader.xsd">
    <BarcodeReaderDevice DeviceID="f">
        <Name>Barcode1</Name>
        <Instance>1</Instance>
        <BarcodeFilter>
            <Index>0</Index>
            <Offset>0</Offset>
            <Filter>1234@@45</Filter>
        </BarcodeFilter>
        <BarcodeFilter>
            <Index>1</Index>
            <Offset>0</Offset>
            <Filter>Bill%%12345@@</Filter>
        </BarcodeFilter>
    </BarcodeReaderDevice>
</BarcodeReaderClass>
```

Show / Hide Mouse Pointer

Whether the mouse pointer is shown or hidden in the Application Core window is configurable using a registry setting, and is automatically updated on installation of the Tools component.

By default, the mouse pointer will be hidden, unless the Tools component is installed when the registry key is automatically updated to show the mouse pointer.

Note: This registry key is independent of the icon setting of the operating system.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

The registry key is as follows:

```
HKLM\SOFTWARE\NCR\Advance NDC\ScreenDisplay\ShowMousePointer
```

Valid values are as follows:

- 0 to hide the mouse pointer. This is the default. (Runtime installation)
- 1 to show the mouse pointer. This is automatically set on installation of the Tools component. (Development PC installation)

Displaying Additional Diagnostics Menus

Where there is no access to VDM and VDA diagnostic support, you can choose to display additional diagnostics menus within Supervisor. These menus allow the display of information on the fitness of present and installed XFS devices. You can then perform a reset, and in some cases a self test, where required.

By default these menus are not shown, and can only be displayed by updating the following registry key:

```
HKLM\SOFTWARE\NCR\Advance
NDC\Supervisor\Diagnostics\MVDiagnostics
```

Valid values are as follows:

- 0 to hide the diagnostic menus. This is the default.
- 1 to display the diagnostic menus.

When operator intervention is required in the diagnostics, a timeout is applied. For example, following the reset of the card reader, an operator must enter a card. If no card is entered before the specified timeout period, the test times out. The timeout period is set in the following key:

```
HKLM\SOFTWARE\NCR\Advance NDC\Supervisor\Diagnostics\Timeout
```

Valid values are entered in seconds. The default is 7 seconds.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Silent Debug

This section describes the installation and configuration of the Silent Debug tool for debugging in a live environment. For information on using Silent Debug, see “Troubleshooting with Silent Debug” on page 11-9.

Silent Debug Installation

Silent Debug is installed as part of the Advance NDC aggregate in a live environment with tracing disabled and as part of a development installation.

Advance NDC installs the *NCRSilentDebug.exe* file in the *C:\SSDS\DLL* directory and the *SilentDebug.ini* configuration file in the *C:\Program Files\NCR APTRA\Advance NDC\Config* directory. If the configuration file is not present in this directory the configuration is not recognised as valid, and the following message is logged to the Windows Event Log:

```
The NDCSILENTDEBUG service configuration file is not valid
```

To overcome potential power failures when tracing in a live environment, Advance NDC installs Silent Debug as an operating system service in Automatic Reset Mode. This can be changed to use Manual Reset Mode as follows:

1. Select **Start | Control Panel |Administrative Tools |Service.**
2. Select **NCRSilentDebug** from the list of services.
3. Select **Action | Properties.**
4. Select **Manual** from the **Startup type** drop down list.
5. Select the **OK** button.

Configuring Silent Debug

The configuration settings for the Silent Debug can be edited in the *SilentDebug.ini* file which is stored in the following location:

```
C:\Program Files\NCR APTRA\Advance NDC\Config
```

If the configuration parameters are not within the permitted ranges, or the *SilentDebug.ini* file cannot be validated, Silent Debug will not start logging.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Configuration options in the *SilentDebug.ini* file are described in Table 5-21.

Table 5-21
Silent Debug Configuration Parameters

Parameter	Description
MAXFILESIZE	Defines the maximum permitted size of the log files in kilobytes (KB). The file is overwritten from the beginning when the file size exceeds the specified value The minimum setting is 1. If MAXFILESIZE is set to zero, Silent Debug will not start. By default, MAXFILESIZE is set to 10 MB A counter keeps track of the number of times the file is overwritten. The counter starts at zero and increments by one each time the file starts to overwrite the log trace data. In this way, old trace data will gradually be overwritten with new trace data. An <END> marker is placed at the end of the last input of new trace data to the file. If the counter indicates a high incidence of overwriting in a short space of time, increase the value of MAXFILESIZE. Sufficient disk space must be available for logging and for other purposes, such as EJ backup files or PD investigations.
TRACE	Gets the log names for each trace stream that is to be logged. The default log names are listed in Table 11-1 “Default Trace Stream Types” on page 11-5. Log names are separated by a comma. You can also add names representing additional trace streams. For further information *see* “Advance NDC Trace Information” on page 11-5.
MERGEDTRACE	Defines whether trace information for each log name defined in the TRACE parameter is captured in individual files, in a merged file, or both. Valid values are as follows: ● 0 - an individual file is created for each log name ● 1 - an individual file is created for each log name and an additional file is created that holds merged trace information for all of the log names. ● 2 - one file is created that holds merged trace information for all of the log names. This is the default. The merged trace file name will take the form *MergedTrace.log* or *MergedTrace_<dd>.log* if daily files are specified.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Parameter	Description
LOGBYDAY *See* Table Note 6	Configures when new log files are created. Valid values are as follows: ● 0 - the same log files are used until they reach the value specified by the MAXFILESIZE parameter, after which the log files are overwritten ● 1 - a new log file is created for each specified trace as specified by the MERGEDTRACE parameter on every day of the month that Silent Debug is running. This is the default If configured to use daily log files, logging to the current files ends at 12 midnight. Logging is then started to a new daily file, which is created in one of the following ways: ● If no files exist for the date, new log files are created as defined in the configuration ● If files exist for the date, the existing log files are overwritten. Daily log files contain the trace data type and day of the month in their file name. For example, if a trace log is created for the MESSAGEIN trace on the second day of the month, the name of the log file is *MESSAGEIN_02.log*.

Table Note 6: If an exception occurs and Silent Debug is running, it will restart automatically when the SST reboots. If Silent Debug automatically restarts and LOGBYDAY is set to 1, logging will continue as follows:

- If the date is the same as when the exception occurred, logging continues in the same file
- If the exception occurs just before midnight and the SST reboots after midnight, logging continues in a new log file.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

DebugLog

This section describes how to install and configure the DebugLog tool for debugging in a test environment. For information on using DebugLog see “Troubleshooting with DebugLog” on page 11-13

Note: The DebugLog tool is supported from Advance NDC 3.02. DebugLog was available in previous releases of Advance NDC but was not supported. It is designed for use in a test environment where a mouse and keyboard are available.

DebugLog Installation

DebugLog is installed with the Advance NDC Tools component in the development environment. For information on installing the Tools component on an SST see “Tools Component” on page 10-19.

Configuring DebugLog

After installation, DebugLog can be configured using the registry. Both registry keys are DWORD types.

Whether to log debug information is set using the following registry key:

```
HKEY_LOCAL_MACHINE\SOFTWARE\NCR APTRA\Advance NDC\
DebugLog
```

Valid values are as follows:

- 0 - no debug information is sent to DebugLog
- 1 - debug information is sent to DebugLog if the utility is active. This is the default

Configuring Auto Save

By default DebugLog automatically saves data to log files for the default trace streams. The files are saved to a directory called *AUTOSAVE* created during the DebugLog installation.

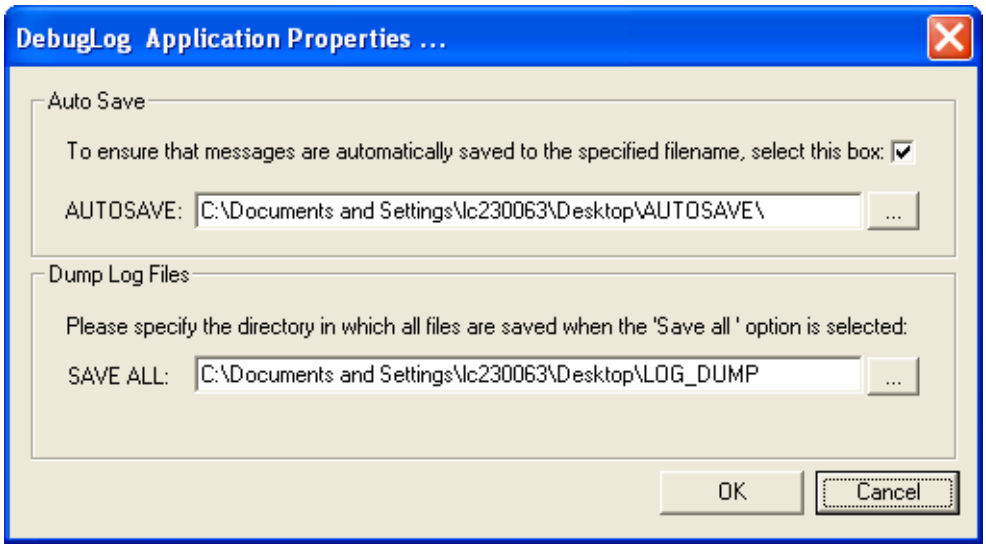
The auto save function can be configured as follows:

1 Open the DebugLog application window.

2 Select **Options | File Options.** The DebugLog Application Properties dialog box is displayed as illustrated in Figure 5-4.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Figure 5-4
DebugLog Application Properties
Dialog Box



3 Select the checkbox to turn auto save on, or deselect it turn auto save off.

4 You can specify a different location for the auto save log files using one of the following methods:
   — Type the required path in the AUTOSAVE text box.
   — Select the **Browse** button at the right of the AUTOSAVE text box, browse to the desired directory and select **OK**.

5 Select **OK** in the Debug Application Properties dialog box.

Changing the Save All Destination

By default, all trace data is saved to the *LOG_DUMP* directory when **Save All** is selected from the DebugLog **File** menu. The destination can be changed as follows:

1 Open the DebugLog application window.

2 Select **Options | File Options.** The DebugLog Application Properties dialog box is displayed as illustrated in Figure 5-4.

3 Specify a different location for the log files using one of the following methods:
   — Type the required path in the SAVE ALL text box.
   — Select the **Browse** button at the right of the SAVE ALL text box, browse to the desired directory and select **OK**.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

4 Select **OK** in the Debug Application Properties dialog box.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

The *Custdat.exe* Utility

Custdat.exe is a command line utility that allows access to the customisation data database, which by default is *custom.dat*. The customisation data database stores the download data sent from the host.

You can use the *custdat.exe* utility to do the following:

- Export the *custom.dat* file

  Once exported, the data can be viewed, analysed, or used in a test environment.

- Import the *custom.dat* file

  The data can be imported to modify the content of the *custom.dat* file, for example, to set up test environments. Do not use the import to adjust a live environment. Changes to a live environment should be made using downloads or local customisation data.

The Customisation Data Database File

The *custom.dat* file is created by Advance NDC after the host sends the following in order:

1 Customisation data download

2 Send Configuration ID message

3 Go In Service command.

The *custom.dat* file contains the following:

- Screens and keyboards. This includes all screens and keyboards downloaded by the host, including any downloaded reserved screens.

  Note: Reserved screens that are not downloaded by the host are stored in the local *resrvd.def* file. The screens in the *resrvd.def* file are not copied to the *custom.dat* file.

  Note: Although keyboards can be downloaded by the host, Advance NDC does not support these keyboards.

- States. This includes all state definitions downloaded from the host.
- FITs. This includes all FIT definitions downloaded from the host.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Using the Custdat.exe Utility

To start the *custdat.exe* utility, enter `custdat`, or `custdat.exe` at the command line alone or with appropriate options, as described in "Exporting" and "Importing".

When the utility is run without an option or with the `-help` or `-?` options, a help screen is displayed.

Exporting

The custom database file can be exported using the following command:

```
custdat custom_file
```

This exports the custom database file, *custom_file*, to the standard output. By default standard output is sent to the screen, but it can be redirected to a specified file. For an example of redirecting the standard output, see "Error Messages" on page 5-74.

Further options can be used, as shown in Table 5-22. If options are used, the command is as follows:

```
custdat [options] custom_file
```

Table 5-22
Custdat.exe Export Command Options

Option	Description
-export	The optional export command
export_file	The name of the custom database file to be exported to the standard output. If this is not specified, the data is exported to standard output by default.
-merge *N*	The optional merge command to specify the number of screens, states, or FITs to include in each message. The default is 1, which means that each screen, state, or FIT is transformed into a single message. This option can be used with either format.
-size *S*	The optional size command to specify the message size. The default size is 600 characters. This option can only be used with the NDChost format. The maximum message size supported by the ndchost format is 9999 characters.
-ndc	Sets the export format to match that used by the host. This is the default format. The length of the line is not restricted when using this format, so the `-size` option has no effect. The exported data contains comment lines, which start with the '#' character.
-ndchost	Sets the export format to match that used by the NDCHost Emulator tool, which is supplied with Advance NDC.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Importing

The file can be imported using the following command:

```
custdat -import import_file custom_file
```

This command would import the file called *import_file* to the custom database file, *custom_file*.

Only the NDC format is supported when importing to the custom database file.

Error Messages

Error messages are sent to the standard error stream. By default, this is the screen.

If you want to capture error messages in a file along with the standard output from an export, enter the following command:

```
custdat custom.dat >log.txt 2>&1
```

In this example, standard output would be redirected from the screen (using the '>' command) to the *log.txt* file, and the error messages (identified by the '2') are also redirected (again using the '>' command) to the *log.txt* file (identified by the '&1').

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Chapter 6

Introducing the Advance NDC Authored Applications

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Overview

Read this chapter if you want to familiarise yourself with the Advance NDC application flow in the Author (the Customisation Layer, Application Core and Supervisor applications).

"Supervisor Mode" on page 6-26 also includes an overview of the separate Supervisor application. Before Advance NDC 2.05, this functionality was provided as part of the Application Core.

Information on how to test the applications on a development system is then provided in "Testing the Advance NDC Application" on page 6-32.

If you simply want to migrate your NDC+ application to Advance NDC and use your existing (unchanged) NDC+ download with Windows XP, see Chapter 3, "Migrating Existing NDC+ Applications to Advance NDC".

If you want to upgrade from a previous release of Advance NDC, see Chapter 4, "Upgrading from Earlier Releases of Advance NDC".

If you intend enhancing the Advance NDC applications, read this chapter first, and then read Chapter 7, "Enhancing the Customisation Layer".

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

The APTRA Author

As described in Chapter 1, “Introducing Advance NDC”, the Customisation Layer, Application Core and Supervisor are authored applications. If you want to view or edit them, you need to know how to use the APTRA Author. If you have the NCR simulator installed, you can use it to simulate devices on the development PC.

To learn how to use the Author and the NCR simulator, refer to the following:

- Author - *APTRA Author User’s Guide*
- NCR simulator - *APTRA Simulator, On-line Documentation*

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Customisation Layer

Before you can enhance the Customisation Layer, you need to understand how the application is structured.

The Customisation Layer is provided within an APTRA Author project. The project file is named *CustomisationLayer.mpj* and can be found in your *<global> \ final \ support \ user* directory.

To access the supplied project file, start the Author and open the *CustomisationLayer.mpj* project.

The project contains several catalogs (under **File | Open Catalog...**). The main Advance NDC catalogs are as follows:

- Customisation Layer Applications
- CDI - <name>
- Customisation Layer
- NDC Core
- NDC Transactions
- NDC Encryption Keys
- NDC Field Workers
- User-Defined CDI.

We discuss each of these catalogs in the following subsections.

Customisation Layer Applications Catalog

The Customisation Layer Applications catalog contains the Customisation Layer application, which uses the following key NDC-specific workers

- Session Requester and Session Releaser - synchronise the Application Core and Customisation Layer
- NDC Processing State Executor and NDC Close State Executor - execute NDC+ States as a black box
- Transaction Request State - sends Transaction Request messages to and receives Transaction Reply or Interactive Transaction Response messages from Central.

Before reading on, open a view of the Customisation Layer application by doing the following:

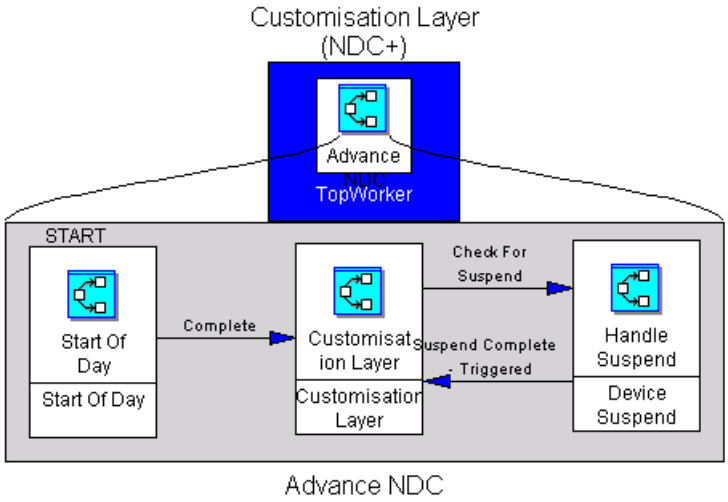
1. Open the Customisation Layer Applications catalog.
2. Double-click the Customisation Layer application.
3. Double-click the Advance NDC Director.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

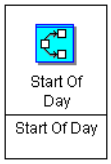
By browsing the application in the Author at the same time as reading this section, you will make the most of the information. You can also browse the Component Descriptions and On-line Help in the Author.

Look at the following diagram to familiarise yourself with the example of the Advance NDC Director in the Customisation Layer application:

Figure 6-1
Example of the Advance NDC Director in the Customisation Layer



Each of the work groups in the Advance NDC Director perform the following jobs:



Start Of Day

This work group displays the Start Of Day message, and waits for confirmation that the Application Core has been started. Once the Application Core has been started, the appropriate journal configuration is then set up, logging of device status information is enabled, and the communications connection is opened.

Status information is logged for the following devices: Statement Printer, Receipt Printer, Cash Handler, Envelope Depository, Envelope Dispenser, Journal Printer, Encryptor, Card Reader, Front Keyboard, Bunch Note Acceptor (BNA) and Cheque Processing Module (CPM).

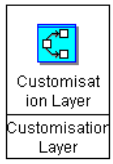
Note: The receipt printer is initialised and prints a test receipt. This pushes any receipt that has not been removed out. If the test receipt is not removed, it will be retracted where the hardware has the capability.

Journal Configuration Journal mode is persistent. At Start of Day, the Configure Journal process checks whether a physical journal printer is present or not. If a printer is found, the previous journal

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

mode is retrieved from persistent storage (*pmdata* file) and used. If a printer is not found, persistent storage is checked for the previous mode setting. If the previous mode was EJ, the journal mode is configured for electronic journalling. If the previous mode was not EJ, the journal mode is set to NONE.

Note: At initial boot-up after installation of Advance NDC, the journal mode is set to PAPER. If persistent storage is unavailable (*pmdata* is corrupt or has been deleted), the journal mode will default to PAPER or NONE depending on whether a journal printer is present or not.



Customisation Layer

By opening the Customisation Layer Director, you can view the Wait For In Service, Session and Close work groups.

Wait For In Service In this work group, when the Application Core enters In Service Mode, a Session Requester worker signals that the session request has been granted.

A cardholder session can then be offered. When a cardholder session has completed, the Session Requester requests that another session be initiated. If a mode change has occurred in the Application Core, the request is denied, and the SST appears Out Of Service to the cardholder. The Session Requester waits until a new session can be initiated.

Note: The Session Requester and Session Releaser workers provide the interface and synchronisation between the Customisation Layer and Application Core. The Session Releaser is described in the following subsection and in “Synchronising with the Customisation Layer” on page 6-16.

Session The Session work group performs the following:

- Executes your previously downloaded NDC+ State Flow using the NDC Processing State Executor worker. This worker executes NDC+ States after the cardholder enters a card (that is, after State Number 0) and before the Transaction Request or Close States
- When a Transaction Request State is detected, the Transaction Request State worker builds and sends a Transaction Request message to the Central application (such as, authorise a cash dispense) and receives a response (which could be to dispense cash)
- After a transaction has been carried out, the rest of the State Flow is executed by the NDC Processing State Executor

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

- The Customisation Layer retains control until a Close State is reached, at which point the cardholder session has been completed.

Within the Session work group, the Card Read work group performs the NDC+ Card Read 'A' or Card Read PIN Entry Initiation 'T' State (these States have been authored in the APTRA Author):

- Checks if the journal configuration has changed since the last transaction. If it has changed, the appropriate journal configuration is activated
- Clears CDI data from the previous cardholder session
- Enables the Card Reader device and prompts the cardholder to enter a card.

  While waiting for a card to be entered, a Session Releaser worker monitors for mode changes (such as, go Out Of Service) occurring in the Application Core. The Session Releaser is used to give up control of the Customisation Layer to the Application Core. However, control is only relinquished when the cardholder session is either idle, waiting for card entry, or at another suitable point in the application when it is safe to return to Out Of Service Mode.

- If a cardholder session is initiated by a cardholder entering their card into the reader, the card is read and a Financial Institution Table (FIT) search is performed

  Any mode changes in the Application Core are delayed until the current cardholder session is completed. The session is completed when a Close State is reached. Upon completion of a cardholder session, another session must be requested, so control is passed back to the Wait For In Service work group.

- If State 'T' is executed, the cardholder is also prompted to enter their Personal Identification Number (PIN)
- Performs any error recovery (for example, if the card read fails, the card needs to be ejected/captured)
- Sends any unsolicited Device Fault Status messages to the Central application.

Within the Session work group, the NDC States work group executes your NDC States.

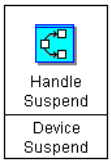
Within the Session work group, the Authored State Types work group executes Advance NDC authored states.

Close This work group performs the following:

- Clears media from the Receipt Printer, BNA, CPM, Card Reader and Statement Printer devices if required

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and/or distribution is strictly prohibited.

- Sends any unsolicited Device Fault Status messages regarding device activity, since completing the last transaction, to the Central application
- Sets up the Next State Number to execute.



Device Suspend

This work group performs the following:

- Handles any suspend condition which may arise with the devices used by the terminal (for example, customer tampering). This effectively puts the Customisation Layer out of service before attempting to clear the suspend condition, ready for a new cardholder session
- Displays the Out Of Service screen until either a Go In Service command is received or five minutes have elapsed, when the suspend condition is cleared.

CDI - <*name*> Catalog

The CDI - <*name*> catalog contains Common Data Interface (CDI) stores. These stores represent items of data that NDC+ makes available as shared data (such as, configuration options and transaction related buffers). For details of the CDI stores we provide, see Appendix C, “Common Data Interface Stores”.

Customisation Layer Catalog

This catalog contains Advance NDC workers specific to the Customisation Layer.

NDC Core Catalog

This catalog contains all the Advance NDC workers common to both the Customisation Layer and Application Core.

NDC Transactions Catalog

This catalog contains all authored functions for the Transaction Request State and NDC Transaction Handler workers to execute a Transaction Reply function.

The workers in this catalog are used in the Tr Reply Function work group of the Transaction Request State and NDC Transaction Handler workers. For details of the NDC Transaction Handler worker, see “NDC Transaction Handler” on page 7-18 and Appendix B, “Advance NDC Workers Supplied”

NDC Encryption Keys Catalog

This catalog contains the Encryption Keys used by NDC+ for Security purposes. These keys are used for PIN Verification and calculating the MAC (Message Authentication Code) in a message.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

NDC Field Workers Catalog

This catalog contains the NDC Field workers which are used to specify the different optional fields that can be included in a Transaction Request message to Central. These workers would be included in the NDC Field work group of an NDC Transaction Handler worker.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Application Core

The main purpose of the Application Core is to preserve the NDC+ message interface and separate the fixed part of NDC+ (the Application Core) from the customisable part (the Customisation Layer, States and Screens).

The Application Core deals with the tasks of mode handling, synchronising the Application Core and Customisation Layer, and handling messages from Central. The remaining sections in this chapter describe how these functions are authored in the Advance NDC application.

Note: The Supervisor Mode functionality is now provided in a separate Supervisor application; for details, *see* "Supervisor Mode" on page 6-26.

The sections are designed to be read in conjunction with the Application Core project file we supply, *ApplicationCore.mpj,* located in *<global> \ final \ support \ user.*

Firstly, open the *ApplicationCore.mpj* project in the Author. You will notice the Application Core project contains several catalogs. The main Advance NDC catalogs described in the following sections are:

- Application Core
- Application Core Applications
- User Messages / Terminal Data Examples
- User Messages / Terminal Data Workers
- User Stores and Signals.

Application Core Applications Catalog

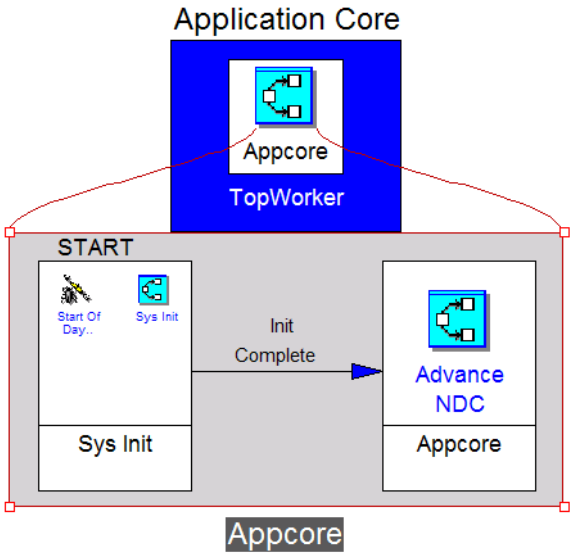
As the name suggests, the Application Core Applications catalog contains the authored Application Core.

Before reading on, open a view of the Application Core application by doing the following:

1. Open the Application Core Applications catalog.
2. Double-click the Application Core application.
3. Double-click the Appcore director and then the Advance NDC director.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

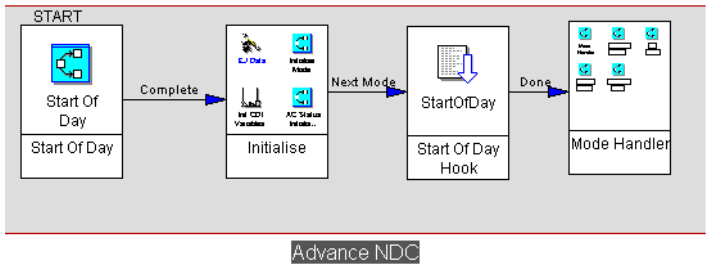
Figure 6-2
Advance NDC Director in the Application Core



By browsing the application in the Author at the same time as reading this section, you will make the most of the information. You can also browse the Component Descriptions and On-line Help in the Author.

Open a view of the Advance NDC director and familiarise yourself with the picture of the Advance NDC Director in the Application Core application:

Figure 6-3
Advance NDC Work Group



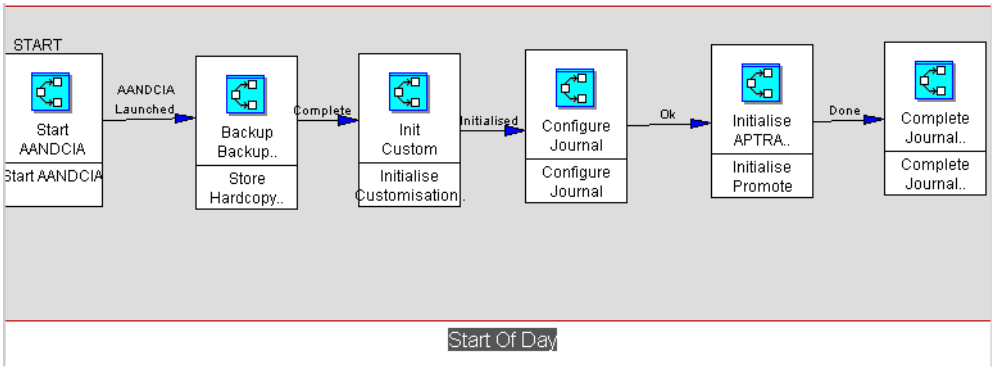
The following sections describe the Start of Day, Initialise and the Mode Handler work groups.

Start of Day

The following screenshot shows the Start of Day work group.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Figure 6-4
Start Of Day Work Group



The Start of Day Director and its subworkers perform the following tasks:

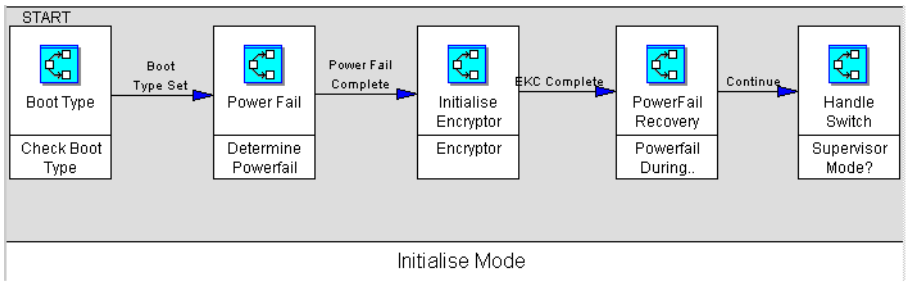
- Initialise Customisation Data work group:
  - Put the In Service Indicators off
  - Initialise or re-load the customisation data from disk according to the Config ID
- Initialise Data:
  - Restore the Transaction Serial Number
- Initialise Devices:
  - Configure the journal printer, including which mode of journal printing is in operation

Initialise

The Initialise work group contains two workers: Initialise Mode and Init CDI.

The Initialise Mode Director performs the following initialisation functions:

Figure 6-5
Initialisation Functions



Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

- Comms if normal boot:
  - — Initialisation of communications with Central (if the mode switch is in the Normal position)
  - — Initialisation of printing on the journal depending on the status of communications
- Determine Powerfail:
  - — System re-initialisation based on the restart mode
  - — Send a Power Up message to Central
- Encryptor:
  - — Re-initialisation of the encryptor if one is present
- Powerfail During Dispense?:
  - — If Powerfail flag set, sets up note counts and performs security trace on last dispense
- Supervisor Mode?:
  - — Checks if the Supervisor switch is in the Supervisor position on start up

The Init CDI script host creates several UCDI stores. For more details of UCDI stores, see Chapter 8, “Enhancing the Application Core or Supervisor”.

Mode Handler

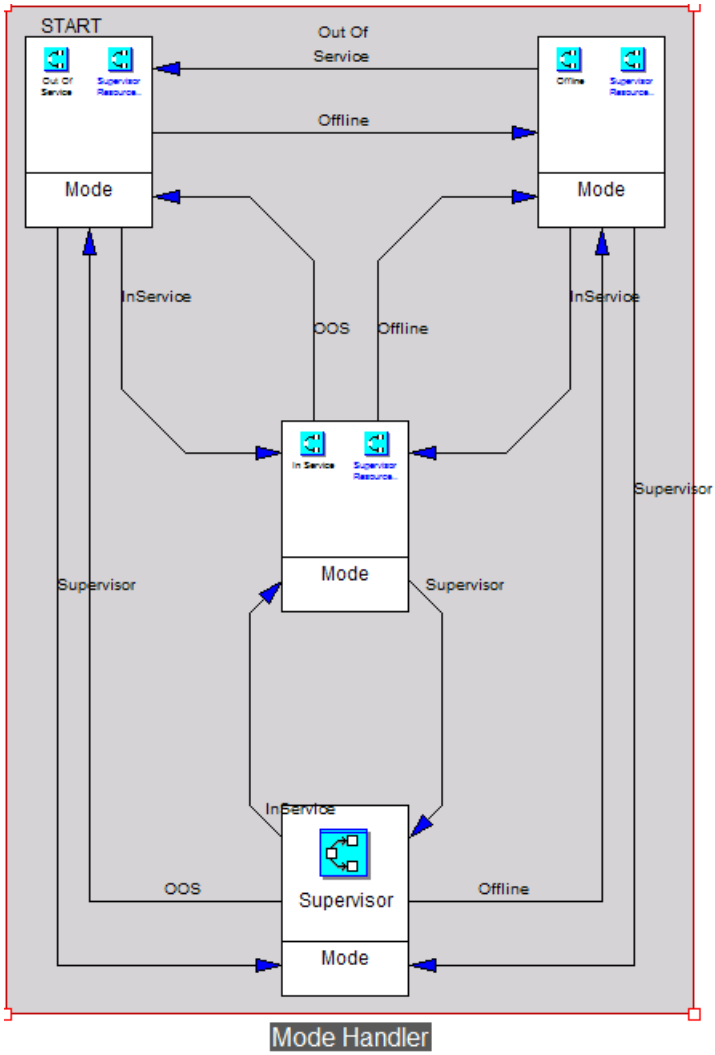
This work group performs the following:

- Handles changes in NDC+ modes (see below for more information)
- Monitors and reports changes in the state of all Tamper Indicating and Alarm sensors
- While the terminal is not in Supervisor Mode, it displays a cycling list of the terminal’s current device-related faults on the Enhanced Operator Panel (if present).

The Mode Handler is authored as a Director with a fixed number of work groups. Each work group is occupied by workers, typically in sub-Directors, to perform the functionality required of each of the NDC+ modes. The work flows between the work groups are used to handle the transitions that occur from mode to mode.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Figure 6-6
Mode Handler Director



Changes between modes are driven by the following factors:

- Terminal Commands (Go In Service, Go Out Of Service)
- Supervisor Mode Switch (Supervisor Entry, Supervisor Exit)
- Option digits, allowing the previous mode to be entered on return from the current mode
- Suspend conditions being detected by the Customisation Layer
- Communications errors
- UPS signals.

Advance NDC will only enter modes based on valid transitions from the current mode. For example, it is only possible to enter Supervisor mode if the mode switch is moved to the Supervisor position and the SST is idle, or the SST is Out of Service awaiting a message.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

The following sections summarise the functionality of each of the modes. For more information about synchronising the Customisation Layer and Application Core, *see* "Synchronising with the Customisation Layer" on page 6-16. For more information about mode and message handling, *see* "Mode Handling" on page 6-17 and "Message Handling" on page 6-19.

In Service Mode In Service Mode is authored as a Director, with subworkers sequenced to perform the required functionality. This worker and its subworkers perform the following:

- Correct handling of messages received with respect to mode of operation
- Synchronisation with Customisation Layer cardholder session activities.

Transitions can be made to the Supervisor Mode, Out Of Service Mode, and Offline Mode work groups.

Out Of Service Mode The Out Of Service Mode is authored as a Director, with subworkers sequenced to perform the required functionality. This worker and its subworkers perform the correct handling of messages received with respect to mode of operation.

Transitions are valid to the Supervisor Mode, Offline Mode and In Service Mode work groups.

Offline Mode The Offline Mode is authored as a Director, with subworkers sequenced to perform the required functionality for Offline Mode. This worker and its subworkers return to correct mode when communications are restored.

Transitions are valid to the Supervisor Mode, In Service Mode and Out of Service Mode work groups.

Supervisor Mode This work group contains subworkers to execute the Supervisor Mode of Advance NDC.

Supervisor mode is implemented in the same way as for NDC+ and includes the following functionality:

- Activation of correct interface on entry to Supervisor Mode
- Correct handling of messages received with respect to mode of operation
- Execution of operator selected Supervisor functions
- Return to correct mode on exit from Supervisor Mode.

Transitions are valid to the Out Of Service Mode, Offline Mode and In Service Mode work groups.

For full details of Supervisor Mode, refer to the *APTRA Advance NDC, Supervisor's Guide.*

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

For details of how to customise Supervisor Mode, see Chapter 7, “Enhancing the Customisation Layer”.

Application Core Catalog

This catalog contains Advance NDC workers specific to the Application Core.

User Catalog

This catalog contains the default versions of the ‘User Messages’ and ‘User Terminal Data’ directors.

User Examples Catalog

This catalog contains an example implementation of the ‘User Messages’ and ‘User Terminal Data’ directors. These examples are described in Chapter 7, “Enhancing the Customisation Layer”.

User Stores and Signals Catalog

This catalog contains the stores and signals (that is, Store workers and Integer Signaller workers) that you are likely to need when updating User Messages or User Terminal Data. These stores and signals should be shared from this catalog when using them in the authored flow.

For details of the stores, see Appendix B, “Advance NDC Workers Supplied”.

The values of the signals are important and must be maintained.

Table 6-1
Signal Values

Signal	Value	When should you use this signal?
Ready9	1	To return a Ready9 terminal state message. Applies only to User Messages.
Reject	2	To return a Command Reject. Applies to both the User Messages and User Terminal Data.
Send Terminal State	3	After setting up the Status Info CDI store with the terminal state message, use this signal to send the message. Applies only to User Messages.
Processing Complete	4	To signal that no reply is to be sent to a new message (class or terminal command) received from Central. Typically used when storing a terminal command. Applies only to User Messages.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Signal	Value	When should you use this signal?
Send Message	5	To send a new message to Central, in response to receiving a new message class. Applies only to User Messages.
TC Processing Complete	1	To send the terminal state message. Any additional data will have already been added to the message. Applies only to User Terminal Data.

Synchronising with the Customisation Layer

The Application Core and Customisation Layer applications are synchronised using two workers - the Session Requester and the Session Releaser.

Session Requester

The main role of the Customisation Layer is to initiate a cardholder session which performs cardholder transactions. In order for this processing to occur the SST must be in In Service mode and ready to initiate a cardholder session. The Application Core synchronises with the Customisation Layer to allow a session to occur. The Session Requester worker in the Customisation Layer waits until the Application Core allows a cardholder session to be initiated.

Session Releaser

When the cardholder session is idle and awaiting card entry, the Application Core can stop the current cardholder session via a Session Releaser worker in the Customisation Layer. If no Session Releaser is active in the Customisation Layer, the Application Core waits until the Customisation Layer has completed its current cardholder session and then denies the request for a new cardholder session.

Checking the Status of the Customisation Layer

The Application Core can query the status of the Customisation Layer during a cardholder session by accessing a CDI Store worker called Customisation Layer Status, found in the 'CDI - Transaction Processing Flags' catalog. This is used to determine if further message processing can take place at various stages in the application execution. This status information is not available to the Customisation Layer directly as it is updated internally by the Session Requester and Session Releaser workers.

The table below shows the possible status values of the Customisation Layer.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Table 6-2
Customisation Layer Status Values

Status	Description
WAITING	Waiting for a Request to be granted.
IDLE	Waiting for a cardholder session to be initiated. The Session Releaser worker is now active, allowing control to be given up by the Customisation Layer.
PROCESSING	Actually performing a cardholder session.
TRANSACTION	Within the Transaction Request State; that is, expecting a Transaction Reply or Interactive Transaction Response (ITR) message. This value is set by the Transaction Handler after a Transaction Request has been sent to Central. It will be set to PROCESSING when the Transaction Handler completes.

Mode Handling

The functionality that handles the changing of modes is called the 'Mode Handler'. The information provided in this section is for information only. **You should not modify the Mode Handler**.

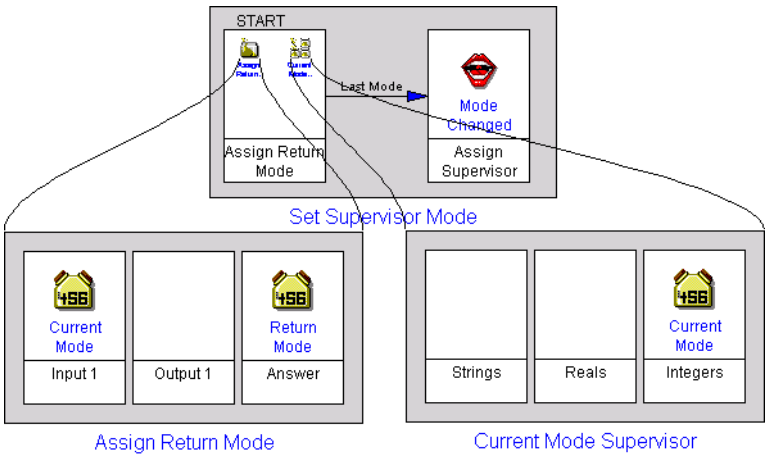
The way in which the Application Core transitions between modes reflects the way in which NDC+ works. Mode changes only occur at specific points in the application, and the way in which a mode change is initiated is similar for each mode.

Caller and Listener pairs are used to detect mode changes. The mode to change to is assigned to the Mode Store and then the Mode Changed Caller is activated. The Mode Changed Listener (below) is concurrent with all activities in the current mode and so will signal when the Mode Changed Caller is activated.

To display the following view from the Mode Handler, double-click Out Of Service | Out of Service *(in OOS Mode)* | Handle Messages.. *(in OOS Mode)* | Message Handler..| Check for Supervisor..*(in Receive Message)* | Set Supervisor.. then Assign Return.. and Current Mode...

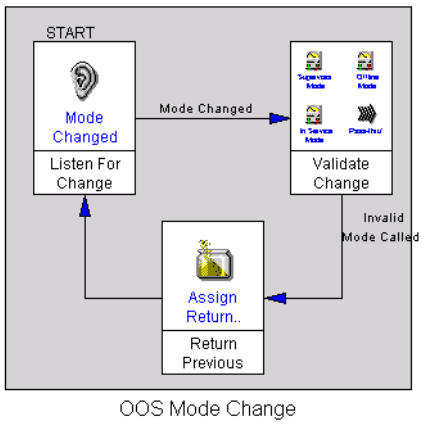
Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Figure 6-7
Mode Handling



To display the following view from the Mode Handler, double-click Out Of Service | OOS Mode Change.

Figure 6-8
OOS Mode Change



The Validate Change work group contains Testers which check whether the current mode change is valid. If it is, a Tester will signal and a mode change will occur. Otherwise, no mode change occurs and the Listener is activated to listen for another mode change. This mechanism is used in each of the modes, allowing the exit to another mode to be initiated from a single exit point.

When a valid mode change is called, all operations in the current mode will be stopped and a mode change will occur. It is important that a mode change should only be called when it is safe to do so. If a mode change should be delayed until an operation has completed then the operation should be synchronised with the mode change. This will ensure that the operation completes before the current mode exits.

Note: You should not modify the Mode Handler functionality. To prevent you from accidentally modifying the Mode Handler

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

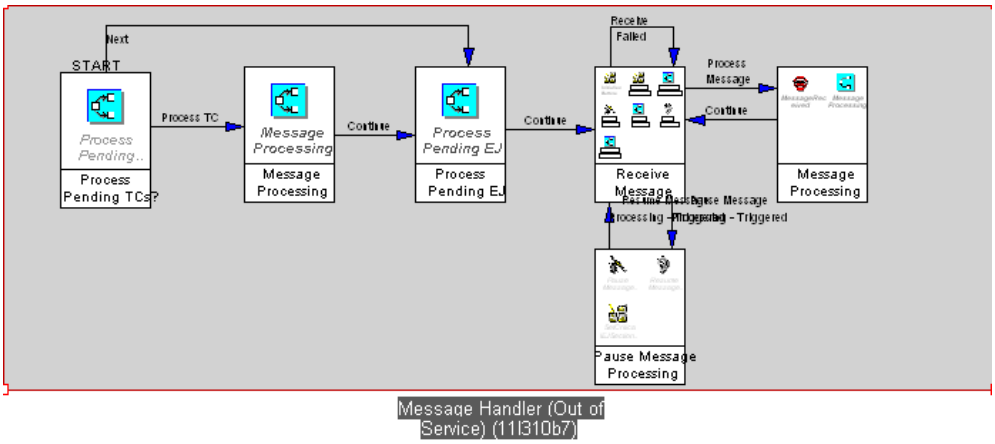
functionality, its workers have been assigned to a different module from those parts of the application you can modify.

Message Handling

Message handling occurs in each of the modes (except for the Offline Mode). Message handling involves receiving a message, processing it and then sending a response (a Ready9, Reject or Terminal State message) to the message. The messages that can be accepted in each mode are clearly defined and are therefore easily processed within each mode. Any message received from Central is processed based on the message class and current operating mode. Message handling has been authored to allow you to extend the SST to Central message interface to process new Message Classes and add additional data to Terminal State messages.

To display the following view from the Mode Handler, double-click **Out Of Service | Out of Service | Handle Messages..| Message Handler...**

Figure 6-9
Message Handling



The Message Handler director handles any incoming messages received from Central. It contains the following work groups:

- Process Pending TCs? - checks for pending Terminal Commands stored in Supervisor and stores the pending commands in the Host Message CDI store
- Message Processing - processes messages from Central and sends appropriate replies. This functionality is discussed in more detail in the following section
- Process Pending EJ - checks for pending EJ upload requests stored in Supervisor and stores pending EJ uploads in the Host Message CDI store

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

- Receive Message - awaits an incoming message, while checking for the mode switch being moved to the Supervisor position

Message Processing Director

The Message Processing Director processes messages from Central and sends appropriate replies. It contains the following main work groups:

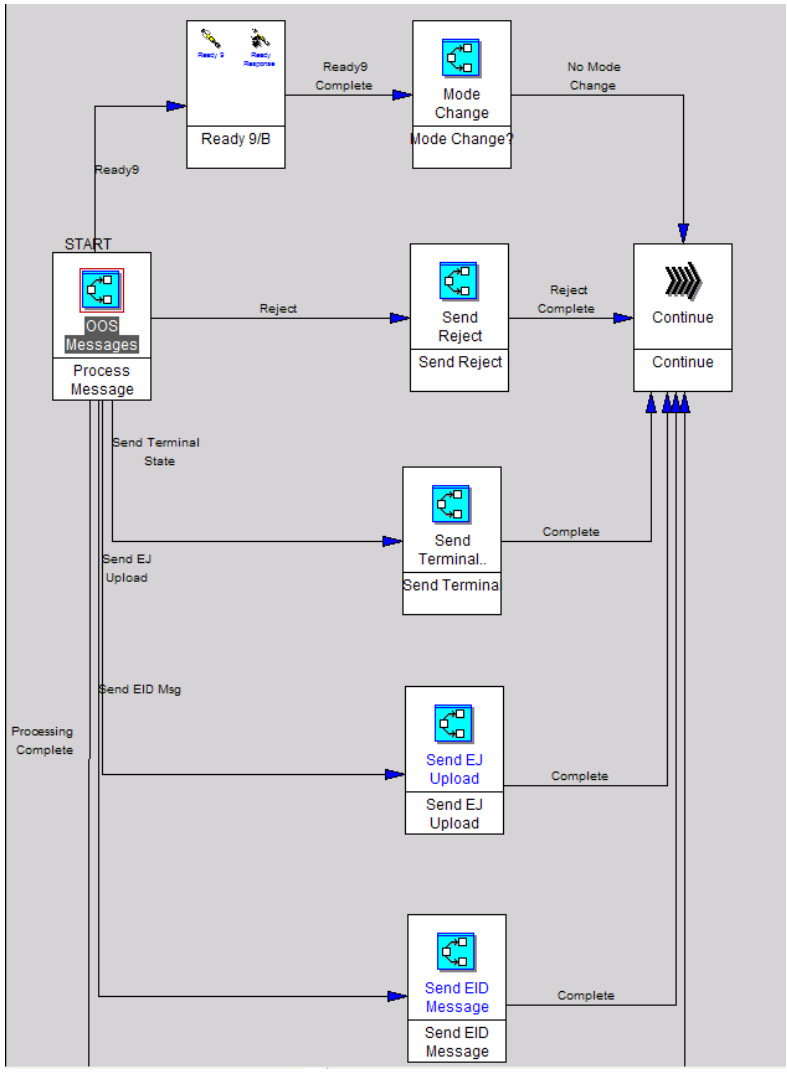
- Process Message - for the particular mode; the next section discusses the work group for Out-of-Service (OOS) Messages as an example
- Ready 9/B - sends a Ready9/B message to Central
- Mode Change? - checks if a mode change is required. If no change is required, the flow goes on to wait for the next message to be received in this mode
- Send Reject - builds and sends a Command Reject or Specific Command Reject Solicited Status Response message to Central, and prints on the journal
- Send Terminal - builds and sends a Terminal State Solicited Status Response message to Central
- Send EJ Upload - checks whether the electronic journal is enabled and assesses the impact of the request before uploading to Central
- Send EID Message - builds and sends an Encryptor Initialisation message to Central.

Note that the Message Processing Director is 'shared' in two work groups. The first occurrence processes any pending Terminal Commands, and the second processes other messages received until the mode is exited.

Process Message The following diagram shows a view of the Process Message workers for OOS Messages.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and/or distribution is strictly prohibited.

Figure 6-10
Process Message



The message is processed by the appropriate work group as follows:

- Message Handler processes messages received from Central. For details, *see* “Message Handler” on page 6-22.
- Ready9, Send Reject or Send Terminal.. Pass-thru workers - when a Ready9, Reject or Terminal State message is to be sent to Central, a signal is sent from the Message Handler
- Mode Change? - work group that sets the values for the required mode and sends a signal for a Ready9 message to be sent to Central
- Send EJ Upload or Send EID Msg. Pass-thru workers - when the EJ upload or an Encryptor Initialisation message is to be sent to Central, a signal is sent from the Message Handler
- Continue Pass-thru worker - this signal is required for situations where the Message Handler is not sending a reply to a message (for example, exit execution).

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Message Handler

The Message Handler functionality processes the following messages from Central:

- Terminal Commands
- Customisation Data Commands
- Transaction Replies.

This section describes how the Message Handler processes these messages.

Terminal Command Processing

Terminal Commands (Message Class 1) in Advance NDC are processed based on the command code of the received message. The range for the command code is 1 to '?' (ASCII 0x30 to 0x3F). Any other command code outside this range is processed by the User Messages director.

The table below shows how the Application Core responds to Terminal Command messages, depending on the current mode. The subsequent table defines the abbreviations used.

Table 6-3
Application Core Terminal Command Responses

Command	Power Up	Out of Service	Supervisor	In Service	Suspend
Go In Service	P1	A1	A2	A	S
Go Out of Service	P1	A	A2	A3	A
Config. ID Request	P1	C	C	C	C
Config. Info. Request. *See* **Note 1:**	P1	D	D	D	D
Counters Request	P1	D	P	R	R
Tallies Request	P1	D	P	R	R
Error Log Request	P1	D	P	R	R
Date / Time Request	P1	D	P	R	R
EKC Retrieve Hallmark Key. *See* **Note 2:**	P1	D	P	R	R
FREE JDATA Enable. *See* **Note 2:**	P1	A	A	R	R
Enable Image Dumping. *See* **Note 2:**	P1	A	A	A	A

Note 1: The information returned by a Config. Info. Request depends on the Command Modifier that is used.

Note 2: These commands are not supported in Advance NDC.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Table 6-4
Application Core Response Definitions

Ref.	Response Description
A	Response is Ready 9 status.
A1	Change Mode to In Service. Response is Ready 9 status.
A2	Hold until Supervisor mode exited, then process. Response is Ready 9 status. (If more than one command is received while in Supervisor mode, only the last is recognised. The others are dropped and no response is given.)
A3	Mode change to Out of Service when terminal is idle at the Card Read State. Response is Ready 9 status.
C	Response is a Terminal State Message containing the Config ID.
D	Response is a Terminal State Message containing the requested information.
P	Hold until Supervisor mode exited and then process according to mode entered.
P1	Hold until Power Up mode exited and then process according to mode entered. No check is made for pending messages on entry to Supervisor or Offline modes, so the message will only be processed on entry to Out of Service mode (unless overwritten by a new request received in Supervisor mode).
R	Response is Command Reject or Specific Command Reject. In order to use these requests the terminal should be put Out of Service. This is to avoid the information being updated by card holder activity while the messages are being created.
S	Hold until Suspend Mode exited and then process. Response is Ready 9 status.

While in service, all Terminal Commands, (except for Config ID requests) are held until the terminal is idle, and then processed. Commands are not stacked and a new request overrides the previous request to which no response is sent.

User Messages and User Terminal Data When the Message Handler receives either an unknown message class or a request for specific terminal state information, it sends a signal to activate the appropriate User work group. This enables you to process new Message Classes and add to existing Terminal State messages.

Customisation Data Processing

Customisation Data commands (Message Class 3, Sub-class 1 or 3) are processed in a similar way to Terminal Commands. The commands are only accepted while the SST is Out Of Service or in Supervisor mode.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Messages with sub-class 2 are interactive transaction response (ITR) messages. These are accepted by the terminal only in response to a Transaction Request message, in order to obtain additional information during a cardholder session. An ITR message received while the terminal is in In Service mode and the Custom Layer Status is TRANSACTION, is processed by a Transaction Handler. The response will normally be another Transaction Request message. Otherwise, the message is rejected (Specific Reject Status / Qualifier 'C04'). An ITR message received in Out of Service or Supervisor Mode will be rejected (Specific Reject Status / Qualifier 'C03').

The table below shows how the Application Core responds to Customisation Data Commands depending on the current mode. The subsequent table defines the abbreviations used.

Table 6-5
Application Core Customisation Data Command Responses

Command	Power Up	Out of Service	Supervisor	In Service	Suspend
State Table Load	-	A	A	R	R
Screen / Keyboard Data Load	-	A	A	R	R
Config. Params Load	-	A	A	R	R
FIT Data Load	-	A	A	R	R
Encryption Key Change	-	A	A	A	R
Extended Encryption Key Change	-	A	A	A	R
Config ID Number Load	-	A	A	R	R
Enh. Config. Params Load	-	A	A	R	R
Diebold PIN Info Load	-	A	A	R	R
Date and Time Load	-	A	A	R	R
MAC Field Selection Load	-	A	A	R	R
Dispenser Currency Cassette Mapping Table	-	A	A	R	R
Override Reserved Screen	-	A	A	R	R
Interactive Transaction Response	-	R	R	A1	A1

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Table 6-6
Application Core Response Definitions

Ref.	Response Description
A	Accept for processing. Response is Ready 9 status.
A1	Accept for processing. Response is Transaction Request if the Custom Layer status is 'Transaction', otherwise it is a Command Reject or Specific Command Reject.
R	Response is Command Reject or Specific Command Reject. In order to use these requests the terminal should be put Out of Service.

Transaction Reply Command Processing

Transaction Reply (Message Class 4) handling relies heavily on the processing carried out in the Customisation Layer and should not be altered. The complete Transaction Request / Reply mechanism is encompassed within Transaction Handler workers to protect the ordering of messages and their content from being erroneously changed.

The first table below shows how the Application Core responds to Transaction Reply Commands depending on the current mode. The subsequent table defines the abbreviations used.

Table 6-7
Application Core Transaction Reply Command Response

Command	Power Up	Out of Service	Supervisor	In Service (Tran. Req. State)	In Service (not Tran. Req. State)	Suspend
Print Immediate	-	A	A1	A	R	R
Other	-	R	R	A	R	R

Table 6-8
Application Core Response Definitions

Ref.	Response Description
A	Accept for processing. If completed successfully, response is Ready 9 or Ready B status. If a device error occurs, response is a device fault Solicited Status. If there is a message format error, response is Command Reject or Specific Command Reject.
A1	If mode was entered from Out of Service, hold until Supervisor mode exited (i.e. Out of Service mode re-entered), then process as A. Otherwise, the response is Command Reject or Specific Command Reject. If more than one command is received while in Supervisor mode, only the last is recognised. The others are dropped, and no response is given.
R	Response is Command Reject or Specific Command Reject.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Supervisor Mode

In Advance NDC 2.05, the Supervisor mode functions were moved from the Application Core to a separate APTRA Author project called *Supervisor.mpj*. The Application Core is thus reduced in size, improving the performance of the Completeness Check and Build Final operations.

To enable synchronisation between the Application Core and the new Supervisor application, new Supervisor Common Data Interface (CDI) stores have been created. These are included in the "Supervisor" section in Appendix C, "Common Data Interface Stores". The stores which have changed to shared stores are listed in this chapter in the "Shared Stores" on page 6-31.

The following sections should be read in conjunction with the provided project files:

- *Supervisor.mpj*
- *ApplicationCore.mpj*

These files are in *<global> \ final \ support \ user*.

By browsing the application in the Author at the same time as reading this section, you will make the most of the information. You can also browse the Worker Class Help by right-clicking on any worker and selecting Help.

The following diagrams are for illustration purposes only; you will gain a more complete and up-to-date understanding of the provided Application Core and Supervisor projects by browsing through them yourself.

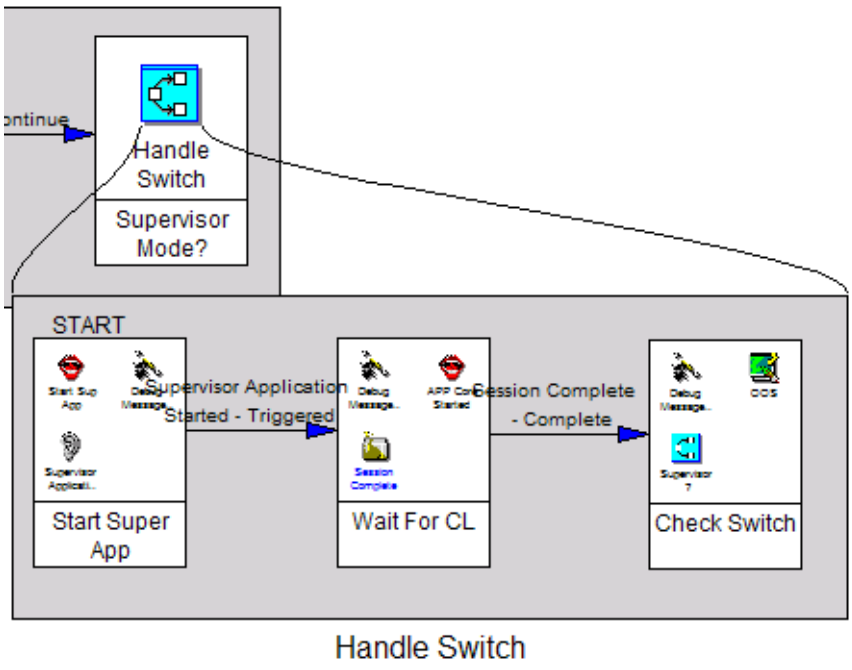
Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Handle Switch Work Group

The following diagram illustrates the Handle Switch work group in the Application Core.

Open the *ApplicationCore.mpj* project in the Author, then select Advance NDC | Initialise Mode | Handle Switch.

Figure 6-11
Application Core Handle Switch



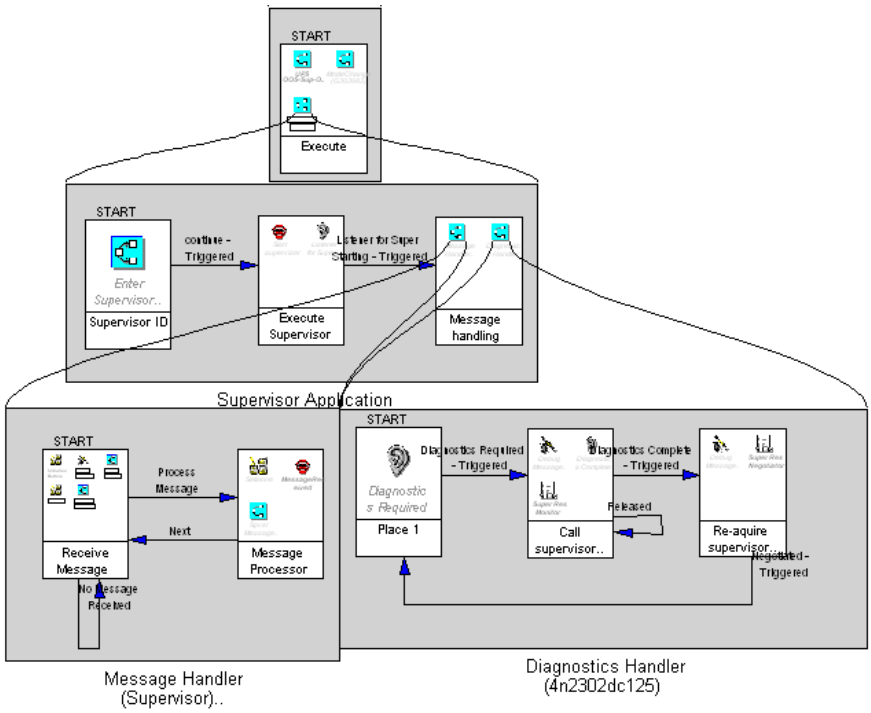
As you can see, the Application Core waits for the separate Supervisor application to start before proceeding.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Starting the Supervisor Application

The following diagram illustrates how the Supervisor Application is started from the Application Core. (*Advance NDC | Mode Handler | Supervisor | Supervisor Mode)*

Figure 6-12
Activating the Supervisor Application



In the Supervisor Application, notice that:

- The Supervisor Application first checks Supervisor Mode is available for entry (Supervisor Present)
- In the Supervisor Application, a Caller / Listener pair is then used to start Supervisor Mode and wait for confirmation that it has started
- Message and Diagnostics handling are started once confirmation that Supervisor Mode has started is received
- The Diagnostics Handler waits for diagnostics to be selected in the Supervisor before acquiring diagnostics and releasing the consumer resource. When diagnostics are complete, the diagnostics are released and the consumer resource is re-acquired.

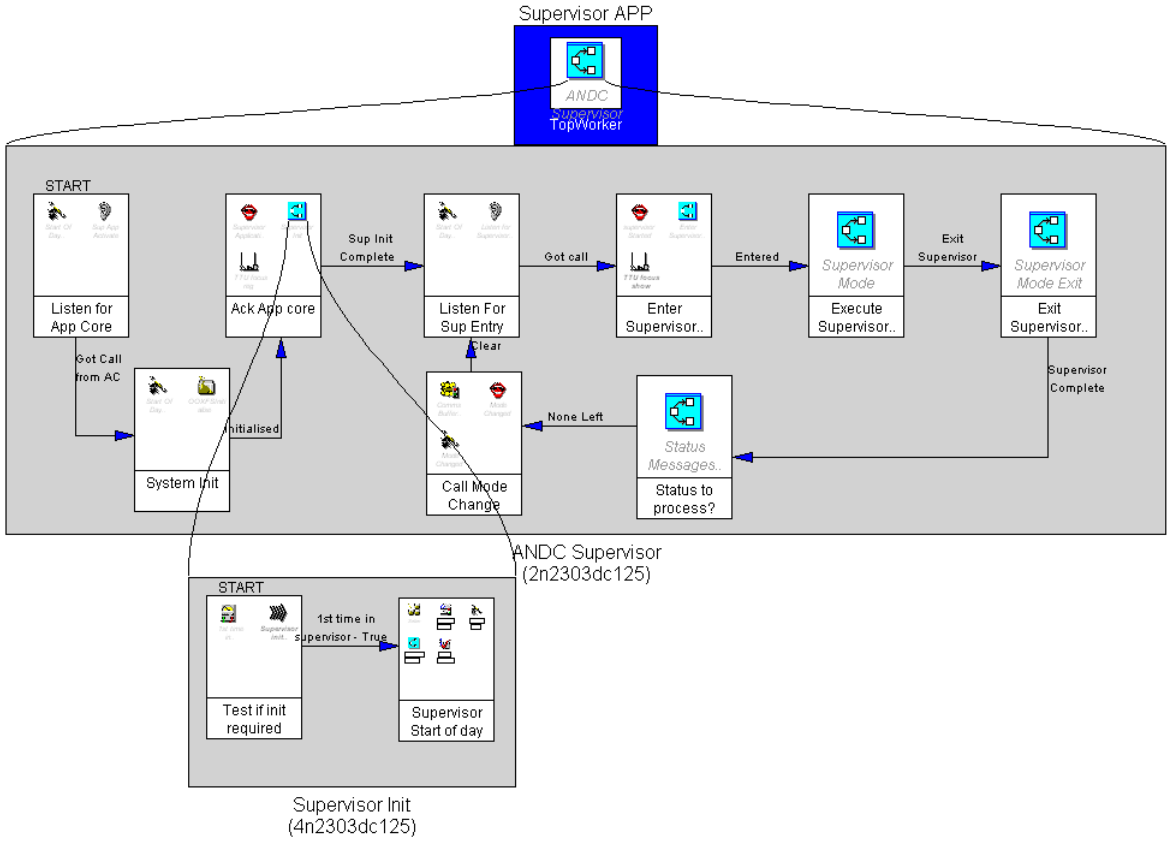
Supervisor Project

The following diagram illustrates the Supervisor Application authored project.

Open the *Supervisor.mpj* project in the Author, and then a view of the Supervisor Application.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Figure 6-13
Supervisor Project



In the Supervisor Application authored project, notice that:

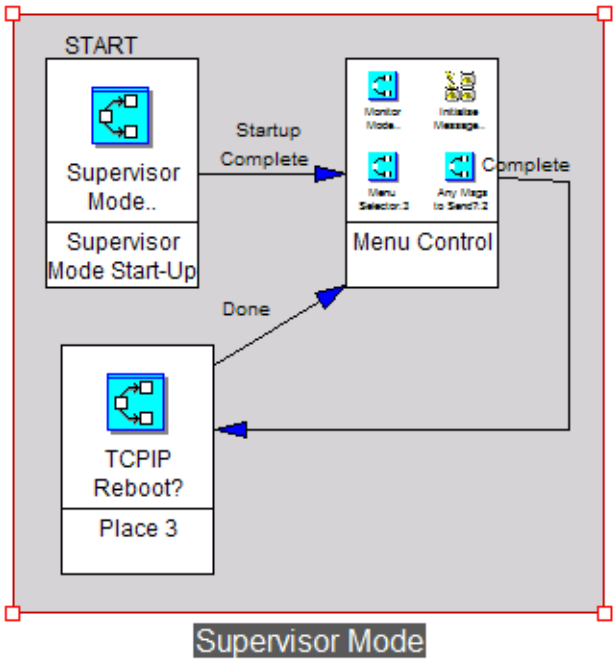
- The Supervisor Application listens for the Application Core to start it
- Once started, the Supervisor Application acknowledges to the Application Core that it has started, using a Caller
- The Supervisor Start of Day then performs any required initialisation, such as setting up communications, and initialising CDI stores
- A further Caller / Listener pair is used in both the Supervisor and Application Core applications to synchronise the entry to Supervisor
- On Supervisor entry, Supervisor Mode is executed (see the following Supervisor Mode Start-Up sub-section)
- On Supervisor exit, the flow returns to listen for Supervisor entry again.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Supervisor Mode Start-Up

The following diagram illustrates the Menu Control in Supervisor Mode Start-Up:

Figure 6-14
Supervisor Mode Start-Up



In Menu Control, notice that:

- No message handling is performed
- Mode monitoring is performed here (previously performed together with the message handling).

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Shared Stores

To enable data to be shared between the Application Core and the Supervisor application, the following stores are shared stores:

Table 6-9
Shared Stores

Store Name	Previous Store Type	Previous Worker ID	Current Store Type	Current Worker ID	Current Shared/CDI Store
Do Not Reset Comms Flag	Integer Store	2r2738r24	Sharable Integer Store	6n2299dc125	Shared
Supervisor Boot	Integer Store	5r2301r22	Sharable Integer Store	18n2299dc125	Shared
Offline Return Mode	Integer Store	5s846g101	Sharable Integer Store	20n2299dc125	Shared
Message Processing Flag (renamed Supervisor Message Processing Flag)	Integer Store	2p8d1	Sharable Integer Store	4r633ichl21	Shared
Supervisor Entry Message (pending)	Integer Store	0n622dc4	Common Data String Store	28r633ichl21	CDI
Supervisor Exit Message (pending)	Integer Store	5n578dc4	Common Data String Store	31r633ichl21	CDI

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Testing the Advance NDC Application

Advance NDC in conjunction with the XFS Simulator allows authored applications to be run in a simulated development environment.

If you have the APTRA Simulator installed, you can use it for testing.

To test the applications, you need to configure all the SST devices you want to simulate on the PC. For details of configuring the communications connection, see “Simulating Communications” on page 6-37.

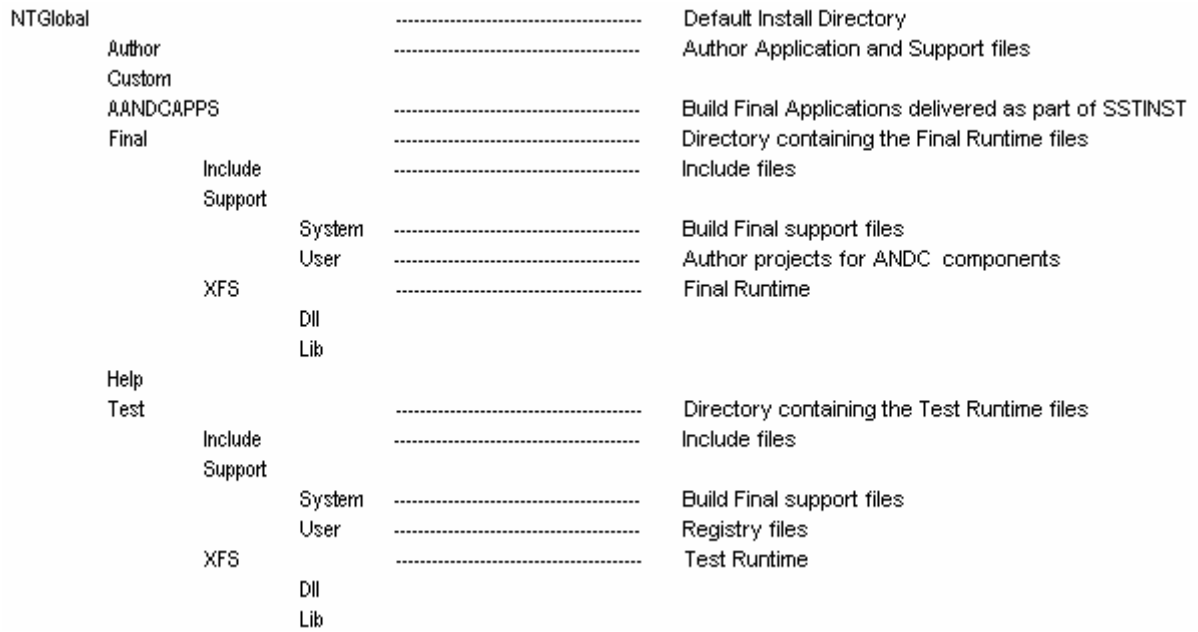
Assuming you are running a test application for the Customisation Layer from the *CustomisationLayer.mpj* Author project, you can run test applications for the Application Core and Supervisor by using SSDS DLL Interface workers to execute previously built Application Core and Supervisor DLLs.

Advance NDC applications can be run on a development PC in two ways:

- Running from within the Author using the start command on an Application icon is called “**running in the Author environment**”. The Author application controls the application context such as the path and exception handling.
- Starting the applications from Windows Explorer or a command prompt is called “**running in the PC environment**”. The application runs in the context of the development PC, including path and exception handling.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Figure 6-15
Advance NDC Test Environment Directory Structure



There are two important environment variables:

%MAPS_GLOBAL%	The directory where the development environment is installed. This is set by the installer.
%ANDC_PATH%	The directory where the runtime will be loaded when an application is executed outside the author environment. The default is *%MAPS_GLOBAL%\final\XFS\dll*. This environment variable is embedded in the PATH environment variable by the development environment installer.

Running in the PC Environment

The applications can be run from Windows Explorer or a command prompt. By default the application windows are full size and cannot be resized. To run the applications in a smaller resizable window, create a system environment variable *APTRA_FINAL_RUNTEST=1*. This will start the applications in a smaller window and all NDC screen displays will be drawn in the top left corner of the screen independently of the applications.

The DLL path used is: *%MAPS_GLOBAL%\final\XFS\Dll*

Running in the Author Environment

Testing within the Authored environment allows developers to 'see' the flow of execution of an application while developing. The Authored environment also gives extended error information and

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

access to help. The application windows can be resized to allow easier access to other applications on the desktop.

To start an application from within the Author:

1 Open the application from the catalog to create a new workspace.

2 Select **Options | Fast Start Enabled**. If this is not selected, the applications will not run in the test environment.

3 Right click on the blue application to select **Start**.

The application initialises and create a resizable window. Depending on what is being displayed in the workspace, red hatching will indicate the currently active work group.

Note: If the application fails to run, a message prompts you to build a worker factory using the Author as the application contains workers that are not contained as part of the default worker factory supplied.

For more information, see the *APTRA Author User's Guide.*

The Author supports the test runtime only, the files referenced are located in *%MAPS_GLOBAL%*`\test\XFS\Dll`, and requires access to libraries in *%MAPS_GLOBAL%*`\test\XFS\Lib` when building a worker factory.

NCR recommends the use of SSDS DLL interface workers to run the applications within the test environment to improve performance when simulating the entire product. Set the SSDS DLL interface workers attribute to the final build DLL of the application under test. For more on SSDS DLL interface workers, refer to the Author online help under *SSDS DLL interface* worker.

Using the XFS Simulator

The XFS Simulator provides full simulation of the Service Providers and hardware and does not require APTRA Self-Service Support to operate. It is only accessible through the CEN-XFS interface.

Preparing the XFS Simulator for Use

When the XFS Simulator is run for the first time it will prompt you to import default ATM and Media definitions. Once these have been imported some adjustments are required in order for the applications to be successfully simulated.

Caution: Applications developed in the test environment must always be tested fully on an SST before being used in a live environment.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Vendor-Dependent Mode (VDM) VDM is not supported on the simulator and Advance NDC will hang if option 7 is selected from the Select menu in Supervisor.

To ensure that Advance NDC offers to close down the PC in this situation and prevent the hang, set the following registry key to 1:

```
HKLM\SOFTWARE\NCR\Advance NDC\VDMSupported
```

If you select 'No' when asked whether you want to close down the PC, you are returned to the Select menu.

Simulated Terminal Text Unit

To ensure the Simulated TextTerminalUnit (TTU) displays the supervisor screens correctly, set the "CR inserts NL" flag on. This setting can be found in the TTU Simulated Device Editor (SDE) by selecting **Capabilities | Standard**. The Simulated TTU service will hang if the ***Enter*** key is pressed without entering any data. It is easier to use the front interface on the development PC for Supervisor. In this case do not include the TTU in the Simulated SST.

Simulated PINpad

The Simulated PINpad must be set to report a different vendor, as the default behaviour for NCR hardware requires further configuration steps to be carried out. This configuration is not required in the simulated environment.

Complete the following steps:

1. Open the SDE for the "Encrypting PINPad EPPB" (PINpad1).
2. Select Responses from the left-hand pane.
3. Open the response details for the **WFS_INF_PIN_CAPABILITIES** command.
4. Click the output button.
5. Change the Vendor key to the value `Other` or any value other than `NCR`.
6. On the **Capabilities | Service Registration** page, change the "vendor name value" to `Other` or any value other than `NCR`.
7. Save the property values.

Ensure that on the PINpad device capabilities the default encryptor state is **WFS_PIN_ENCREADY** before simulation proceeds.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

These updates will take effect when the simulation is closed, then restarted.

Currency Dispenser

The Advance NDC currency mapping table in the registry needs to be updated to reflect the default configuration in the XFS Simulator. Updating the 'CurrencyID' field in each of the NDC Types to USD should ensure that money can be dispensed during the application simulation. See "Cash Handler Configuration" on page 5-2 for more information.

The other alternative is to edit the Cash Units tab in the Currency Dispenser Device Capabilities SDE and change the 'cCurrency' field to GBP.

Coin Dispenser

The simulator must be configured to emulate the coin dispenser, as follows:

1 Select RS232 coin dispenser from the list of devices in the simulatior to include the device.

2 Open the Simulated Device Editor (SDE) for the coin dispenser and select **Device Capabilities | Cash Units** to see the current currency (cCurrencyID) and value (ulValue) settings for a hopper (Chute).

3 Edit the fields as shown in Table 6-10.

Table 6-10
Coin Dispenser Simulator Settings

Chute	cCurrencyID	ulValues
1	USD	10
2	USD	20
3	USD	50
4	USD	100
5	USD	10
6	USD	20
7	USD	50
8	USD	100

4 From the **Capabilities** option on the **Standard Capabilities** tab, select "bItemsTakenSensor" and edit "wMaxDispenseItems" by entering the maximum dispense in the **Unit Limits** text box.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Set up the simulator to generate eight removal or insertion events simultaneously with the RS232 coin dispenser device as follows:

1 Open the Simulator Console, and select Scripting.

2 Select New and name the new script “Remove Canister” or “Inser Canister” as appropriate.

3 Start recording and go to the coin dispenser IDE.

4 Select Device | Failures | Physical Cassette *x* - Chute 1, and then select the following:
   — WFS_CDM_STATCUMISSING for removal
   — WFS_CDM_STATCUOK for insertion.

   Repeat this for each of the hoppers replacing *x* with the hopper type.

5 Stop recording.

6 Update the status of all hoppers from missing to low.

To simulate the removal or insertion, click on the appropriate script with ***MB2***, select Play with the ‘Zero all playback timing delays’ checkbox selected, and select OK.

Simulating Communications

To simulate communications on a development PC, you need to configure Advance NDC and the communications service for TCP/IP. For details, see “Configuring Communications” on page 5-11.

Advance NDC provides an NDC Comms Connection ID worker, which determines the communications used from the registry settings.

Note: As PCCM is proprietary software of NCR, it is not supported by the APTRA Simulator.

The applications can now be run in the test environment, in the following order:

1 Application Core

2 Supervisor

3 Customisation Layer.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and/or distribution is strictly prohibited.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Chapter 7

Enhancing the Customisation Layer

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and/or distribution is strictly prohibited.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Overview

In Advance NDC, you can customise an Advance NDC application by modifying the Customisation Layer, Supervisor and Application Core applications.

This chapter describes how to:

- Make changes to the Customisation Layer. Remember that making changes only to the Customisation Layer means that your application will maintain compatibility with NDC messages.
- Add new data fields to a Transaction Request, and process new Function IDs and Printer Flags in a Transaction Reply; *see* "Transaction Request / Reply" on page 7-16.

Bear in mind that any customisation you make using the Author will have to be reapplied to future releases of and service pack updates for Advance NDC. Therefore, design your customisation so that your method of implementation has minimum impact on the Authored application as supplied. For suggestions on how to do this, see "Impact of Changing the Customisation Layer" on page 7-6.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Testing the Customisation Layer

If you have the APTRA Simulator installed, you can test your changes on the development PC. To test the applications, you need to configure all the SST devices you want to simulate on the PC.

For details of how to configure devices, refer to the documentation accompanying the APTRA Simulator.

For details of how to configure the communications connection to test the Advance NDC application, see “Simulating Communications” on page 6-37.

To simulate receiving messages from Central, you can enter all messages into the host simulator (NDCHost Emulator) and assign names to them. For example, you will need to enter Transaction Reply messages to respond to the Transaction Request message sent by the Customisation Layer.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

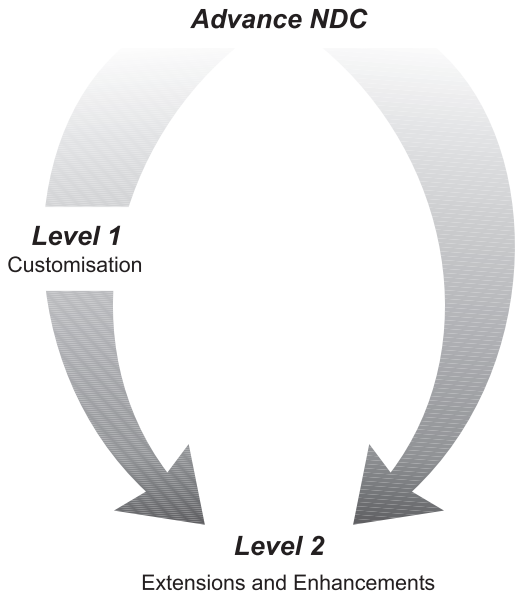
Modification Options

Advance NDC uses both authored workers and NDC+ states and screens and therefore the Customisation Layer can be modified at two levels as follows:

- Level 1 - Customisation using standard NDC+ States and Screens download
- Level 2 - Extending and enhancing using exits and hooks.

The following diagram shows how these modification levels allow you to evolve away from the NDC+ programming model of States and Screens.

Figure 7-1
Modification Levels



Level 1 Customisation

If you want to use Level 1 modifications, for example to tailor the user interface, use the NDC+ customisations such as screen downloads.

For further information on States and Screens, refer to the *APTRA Advance NDC, Reference Manual.*

Level 2 Extensions and Enhancements

Use the Author to include Level 2 modifications, such as the following:

- C exits
- Authored exits

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

- External C functions
- Exit hooks
- Web exits.

For further information, see Appendix A, “Using Exits in Advance NDC” and Appendix G, “Web Exit State”.

Methods of Enhancing the Customisation Layer

If you want to replace all the States and Screens in your Customisation Layer, you can do so by modifying or replacing the cardholder transactions with your own transactions.

If you want to replace some of the State Types with exits or workers or add new authored transactions, modify the Customisation Layer. If desired, you could replace all State Types in your Customisation Layer with exits or worker hierarchies.

Examples of changes you can make to the Customisation Layer are provided in the “Modification Examples” on page 7-10.

Whichever option you decide to take, both levels involve modifying the Customisation Layer application we have provided, to create a customised application.

Note: Before attempting to modify the Customisation Layer, read the following section.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Before Modifying the Customisation Layer

This section provides important information that you should be aware of before modifying the Customisation Layer.

Customisation Guidelines

Your changes to the Customisation Layer must conform to the requirements below, and follow the stages given in the *Preparation Guidelines* section in this chapter.

The conformance requirements we place on you are as follows:

- You follow the protocol we have provided to synchronise the Customisation Layer and Application Core applications. This protocol is known as the Session Request/Release protocol, and is discussed in "Wait For In Service" on page 6-5 and "Synchronising with the Customisation Layer" on page 6-16.
- You save the Customisation Layer Author project (*CustomisationLayer.mpj*) with a different name and back it up, so as to avoid overwriting your changes when re-installing Advance NDC.
- You use a separate directory for each project being worked on, and for each revision of each project. This enables previous revisions to be used or tested if necessary.
- You clearly document the functionality provided in each revision.
- To avoid files being overwritten or deleted during installation of a subsequent Advance NDC release, keep your files in your own working directory, not in the global directory or its subdirectories.
- You do not change or copy application framework workers which must retain their component ID, or which synchronise the applications (Caller workers).
- If you extend the NDC Message Interface by adding new data fields to a Transaction Request or processing new Function IDs and Printer Flags in a Transaction Reply, you need to contact NCR to obtain unique message identifiers. For more details, *see* the following "Compatibility Considerations" on page 7-6.
- Any Transaction Reply Functions must complete immediately on sending a solicited status message. This means that, after sending the solicited status message, only flow control workers and setters should be used.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and/or distribution is strictly prohibited.

Impact of Changing the Customisation Layer

As long as you modify the Customisation Layer we provide according to these guidelines, you can assume it will work with subsequent releases of Advance NDC, but you will have to do one of the following:

- Re-apply your changes to the subsequent release of the Customisation Layer
- Apply the changes required from the subsequent release of the Customisation Layer to your Customisation Layer.

Reducing Future Work

To reduce the time you will have to spend in the future on re-applying your customisations, consider the following:

- Develop your customisation as a separate application and call it as a C Exit, a C function or a DLL Interface worker.
- For Authored State Types, copy the Customisation Layer and use the Replace option down to the level at which you will add the new state.
- Create catalogs for testing / development and workers.
- Make your work as modular as possible, with porting in mind.
- If you use an Authored Exit, apply the design recommendations given in “Portability of Authored Exits” on page A-13.

Compatibility Considerations

If you are extending the NDC Message Interface, you need to allow for future releases of Advance NDC, to remain as compatible with them as possible.

To ensure your extensions are compatible with Advance NDC, you must contact NCR Dundee Product Management to obtain unique message identifiers. Examples of when you need to obtain message identifiers are for a new:

- Message Class
- Command Code
- Field in a message (such as a new data field in a Transaction Request)
- Data Identifier in a message
- Function ID in a Transaction Reply
- Printer Flag in a Transaction Reply
- DIG (Device Identifier Graphic) for a new device.

User IDs

User IDs must be unique, and consist of at least two letters. When using the Author, you must ensure that any user ID used to create new workers does not match NCR’s proprietary user IDs. For details, refer to *APTRA Author, User’s Guide*.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Note: NCR recommends that you use two or three character initials for User IDs, and avoid using numerical characters.

Component IDs

Component IDs must be unique.

You can ensure that the Component IDs remain unique by performing a module copy to merge the new catalogs.

Documenting Changes

Ensure that changes are clearly documented. This facilitates easy review and comparison when making further changes or applying a new release.

Always include a description of the worker. When dealing with flags or stores, it is particularly important that you describe the purpose, location(s) and the range of valid values. Use the Description property of any catalogs, directors and other workers to provide a summary of any update.

Ensure that further documentation includes at least a list of changed workers, with the original component ID and worker name, and a summary of the changes made. Select **Components | Report** to generate a report of updated workers, catalogs, or components in the project. You can export the resulting report into Excel and then filter and sort on the time-stamp and module. This lets you clearly see all the components that have changed in a project during particular sessions. You can also use this file to include additional information to help document the changes.

Note: Including the worker name ensures that you can identify the worker, even after a module copy assigns a new unique Component ID.

NCR recommends that you also produce a document using a package such as Microsoft Word. The document includes the:

- Application being modified
- Catalog being used
- Name and Component ID of the top worker
- Full description of the change and its purpose.

To help clearly communicate the changes in this document:

1 Take screen shots, using ***Alt+Print Screen***

Note: Ensure that important component names are readable in any screen shots and retain some context, such as the upper level workgroups.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

2 Paste the screen shot into your document

3 Add callouts and annotations

Note: If you number the callouts, you can refer to the illustrations in the description of the change.

Documenting Director Changes

Try to keep your changes within directors. This method uses the director as a container for the update. You will also find that this method simplifies the documentation as you only need to replace the director.

Ensure that you include a statement covering whether the change results in a new or updated director.

Documenting Signal Changes

If your changes require signals, ensure that you have identified all that are required. Once you have identified the signals, describe them.

Try to use the logic of the containing director as much as possible to reduce the signals required.

Preparation Guidelines

These guidelines assume you are familiar with using the Author, and you have read the related documentation, and/or attended an Advance NDC or Author course. As such they are not step-by-step instructions, but stages to follow when you are preparing to make the modifications you require.

Before and during modifications, it is good practice to back up the provided projects, and the projects you are working on.

NCR recommends that you follow these guidelines when preparing to modify the Customisation Layer:

1 With the required project file open, create a new module for your development. Change the default module name you are prompted with to an appropriate and meaningful (recognisable) name, based on the customisation you are going to develop.

2 Set your created module as the default module. All workers and catalogs created will be assigned to this module.

3 Create a new catalog ready for your applications, with a meaningful name.

4 Create a new catalog ready to hold your workers, and workers being changed, with another meaningful name.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and/or distribution is strictly prohibited.

Note: If your modifications require new worker class definitions, you should create these in a separate module. This is necessary as a module copy cannot copy a module containing worker classes or literal type definitions.

5 Copy the application to be modified into your applications catalog.

6 Open the application in your applications catalog to display its top-level picture.

7 Before changing a Director worker by adding a worker or work group, share the original worker into your worker catalog. This is useful for future releases or changes, enabling you to review and compare changes.

8 Replace the top worker and any other Director workers you wish to change, down to the required level. Make changes at the Director level wherever possible, to enable easier re-use of your own changes, and for subsequent releases.

9 Ensure all the signals associated with the workers you have changed are reconnected to the required workers.

10 Test the changes you have made in isolation as much as possible, in a Unit test.

11 Perform a module copy on the catalogs containing tested developer(s) changes into another fully functional 'work in progress' application, as and when required.

12 Share the top workers from the catalogs into the main flow.

Note: If changes are made to shared workers, you must ensure that you identify all locations that use the shared workers. Ensure that you have identified all instances of the shared worker, even where the sharing occurs at a higher level.

13 All changes can then be tested in one or more Integration tests, separately from a previously tested (known working) application.

14 New functionality and updates can continue to be added in a controlled manner, using these stage guidelines.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Modification Examples

This section describes how to make the following changes to the Customisation Layer:

- Replace NDC+ Data Entry State Types and Screens with Data Entry and Screen workers to enhance the user interface
- Edit State Types authored in the Customisation Layer, such as the Card Read State, in order to override downloaded customisation data
- Create new State Types.

Examples of how to make these changes are given in the following sections.

Replacing State Types with Workers

You may want to replace NDC+ Data Entry State Types and Screens with Data Entry and Screen workers (which have improved GUI features) to enhance the user interface.

This is possible by exiting the State Flow based on a given State Number and / or State Type, executing a worker hierarchy and then returning to the State Flow based on the Next State Number.

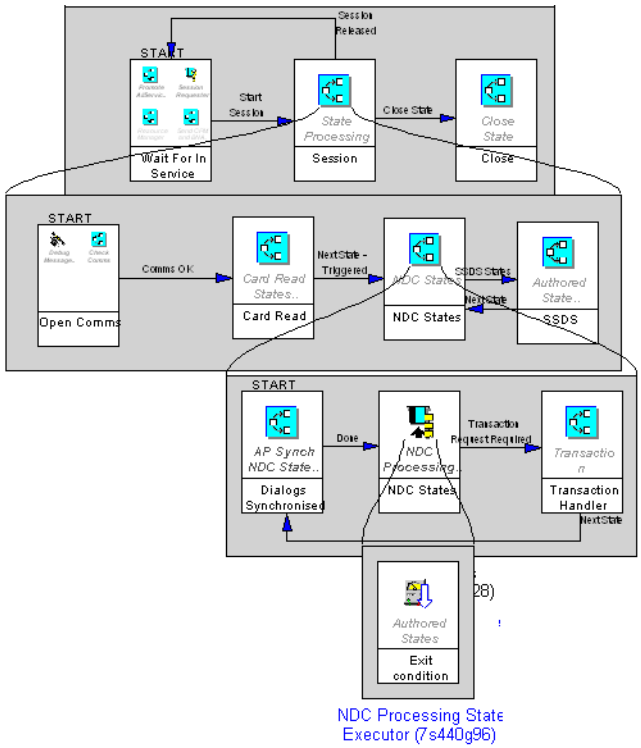
How this is achieved is best explained with the aid of an example. The Customisation Layer application contains an example of exiting a State Flow to execute an authored State Type. We have authored the following State Types in the application:

- PIN Entry ‘B’
- Enhanced PIN Entry ‘M’
- Customer Selectable PIN ‘b’
- Cash Accept ‘>’
- Cheque Accept ‘w’.

Open a view of the Customisation Layer application, as described in “Customisation Layer Applications Catalog” on page 6-3. The following diagram shows the worker hierarchy of the State Processing director.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Figure 7-2
State Processing Director: Worker Hierarchy



Exiting a State Flow

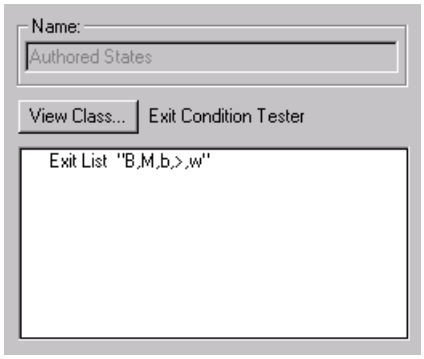
You will notice that the State Processing Director contains an NDC States director and an Authored State Types director. The NDC States director is described in this section and is used to control the application's exit from an NDC State flow to the authored flow and its return. The Authored State Types Director is described in the following section.

The NDC States director contains an NDC Processing State Executor worker which in turn contains an Exit Condition Tester worker. These workers are specific to Advance NDC and are described in Appendix B, "Advance NDC Workers Supplied".

An NDC Processing State Executor executes NDC+ States after State Number 0, the Card Read State, and before the Transaction Request or Close States. It has an 'Exit condition' work group which allows you to exit a State Flow based on a particular State Number and / or State Type, checked for by an Exit Condition Tester worker. The Exit Condition Tester has an Exit List attribute which contains a list of comma separated State Numbers followed by State Types (for example '1,2,3,4,A,B,C'). The following dialog box shows the Exit List attribute for the Exit Condition Tester used in the Customisation Layer application.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Figure 7-3
Exit List Attribute for the Exit Condition Tester



If a State Number/Type matches an entry in the Exit List attribute (that is, ‘B’, ‘M’, ‘b’, ‘>’ or ‘w’), the Exit From Flow True signal (which is tied to the SSDS States work flow in the application) is produced by this Exit Condition Tester, to enable an exit to the authored flow.

Executing a Worker Hierarchy

The Authored State Types director is used to execute State Types ‘B’, ‘M’, ‘b’, ‘>’ and ‘w’, which have been developed in the Author.

When the director is activated by the Exit From Flow True (SSDS States) signal, Author workers are used to determine the Current State Type by interrogating a Common Integer Store worker. The Common Integer Store worker contains the Current State Type CDI data.

Depending on whether the Current State Type is ‘B’, ‘M’, ‘b’, ‘>’ or ‘w’, a Director is activated to execute the appropriate State Type.

Based on the success of the State, the Next State Number Common Integer Store is set, which in turn results in the Next State signal passing control back to the NDC States Director.

Returning to a State Flow

When control is passed back to the NDC States Director, the NDC Processing State Executor resumes executing NDC+ States, from the Next State Number.

The NDC Processing State Executor signals when the next state to be executed is a Transaction Request or Close State.

If a Transaction Request is to be performed, the Transaction Request State worker is signalled. This worker is described in Appendix B, “Advance NDC Workers Supplied”.

If the Close State is reached, the Close State Director is activated.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and/or distribution is strictly prohibited.

Setting the CDI Stores Used in a Transaction Request

When you enhance the Customisation Layer, by replacing NDC+ State Flows with Worker hierarchies, you need to write to the correct CDI workers before using the Transaction Request State worker. For details of CDI stores, see Appendix C, "Common Data Interface Stores".

If you replace all NDC+ States and Screens with Worker hierarchies, you can still perform Transaction Requests. However, you must use the NDC Transaction Handler worker to do Transaction Requests / Replies, instead of the Transaction Request State.

For details of setting the optional NDC Fields used in the NDC Transaction Handler worker, *see* "NDC Transaction Handler" on page 7-18.

Editing State Types Authored in Advance NDC

You can edit the State Types which are authored in Advance NDC, such as the Card Read State.

Instead of implementing the NDC+ Card Read State as a worker class (C++ code) or an external C function, we decided to author it using workers. Therefore, the Customisation Layer is a hybrid of workers and NDC+ States / Screens.

You can edit the Card Read State using the Author. Information such as the screen displayed when prompting for card entry and which tracks are read is taken from the previously downloaded customisation data.

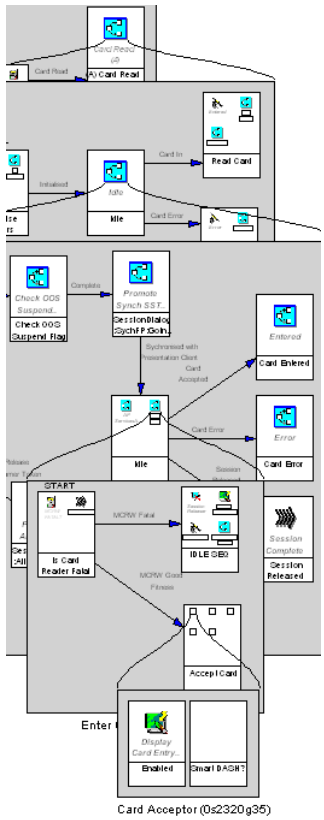
You can override downloaded customisation data by making changes to the workers in the Customisation Layer.

We will now show you how to change the screen displayed when prompting for card entry.

Open a view of the Customisation Layer application, as described in "Customisation Layer Applications Catalog" on page 6-3. The following diagram shows the worker hierarchy of the Card Read States (A)/(T) Director.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Figure 7-4
Card Read States (A)/(T): Worker Hierarchy



The Card Acceptor worker in the worker hierarchy is used to prompt a cardholder to enter a card into the Card Reader.

While the Card Acceptor is active, the NDC Screen Displayer worker, in its Enabled work group, displays a previously downloaded NDC+ screen (that is, the screen specified by Table Entry 2 of the Card Read State).

Workers could be used in the Card Acceptor's Enabled work group to provide a more attractive display sequence.

For example, to display a screen which is not part of the NDC+ download, you could replace the NDC Screen Displayer with a standard Picture Displayer worker.

Since NDC+ can display screens across States, you also need to modify the NDC Close State Executor worker in the Card Read States (A)/(T) Director. This involves setting its Clear Screen On Exit attribute to 'True'. This is necessary to make the screen remain blank until an application screen is displayed (that is, until the Picture Displayer is displayed). If you do not set the attribute to 'True', the application screen will appear behind the last displayed NDC+ screen and hence will be hidden.

Note: If necessary, edit the STCONT Registry file with the changes you have made, as described in "Editing STCONT" on page 3-13.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Creating New State Types

If editing an existing State Type in Advance NDC does not provide the functionality you require, you may wish to create a new State Type in Advance NDC.

If you decide you do need to create a new State Type in Advance NDC, you should use the following guidelines:

1. Consider which State Types you need to add to Advance NDC, and how they are best created / added. In particular, decide whether they should be C Exits or Advance NDC authored State Types.

2. Any necessary new authored State Types should be defined as part of the preparation in “Preparation Guidelines” on page 7-8.

3. Add your new State Type to the Authored State Types selector.

4. If necessary, edit the STCONT Registry file with the changes you have made, as described in “Editing STCONT” on page 3-13.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Transaction Request/Reply

In NDC+, the Transaction Request State sends a Transaction Request message to Central, and executes the Transaction Reply command received from Central. In Advance NDC, Transaction Requests/Replies are performed by a Transaction Handler worker, as part of the Customisation Layer.

We provide two versions of the Transaction Handler worker:

- Transaction Request State - performs the same functionality as the NDC+ Transaction Request State
- NDC Transaction Handler - allows you to specify the fields to include in the Transaction Request message. You only need to use the NDC Transaction Handler worker if you have replaced the NDC+ Transaction Request State with worker hierarchies. For more details, *see* "NDC Transaction Handler" on page 7-18.

Both Transaction Handler workers additionally allow you to add new data fields to a Transaction Request, and process new Function IDs and Printer Flags in a Transaction Reply. This involves extending the NDC Message Interface. See the following sections for details.

If you choose to enhance the Customisation Layer we provide, by replacing NDC+ State Flows with worker hierarchies, you will need to write to the correct CDI store workers before using the Transaction Request/Reply mechanism. For details of CDI stores, see Appendix C, "Common Data Interface Stores".

Before Modifying the Transaction Request/Reply

If you intend extending the NDC Message Interface to add new data fields to a Transaction Request or process new Function IDs and Printer Flags in a Transaction Reply, you need to be mindful of the importance to remain compatible with future releases of Advance NDC. For more details, *see* "Compatibility Considerations" on page 7-6.

Adding Data Fields to a Transaction Request

To allow you to add new data fields to a Transaction Request message, the Transaction Handler workers contain a work group called 'User Data'. You can add new data fields by including String Giver workers in this work group.

The data you supply will be inserted before any MAC (Message Authentication Code) data. Advance NDC automatically precedes the data with a Field Separator character, and each data string is separated by a Group Separator.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and/or distribution is strictly prohibited.

Note: The MAC Field Selection Load message has not been expanded, so Selective MACing of the 'User Data' fields is not possible.

The Transaction Request Extension state was modified to accommodate adding new data fields to the Transaction Request message. See state table entry 6 in the *APTRA Advance NDC, Reference Manual* for details of the changes.

Caution: Ensure you do not exceed the maximum length permitted for the Transaction Request message. The maximum length will depend on the communications protocol being used.

New Function IDs in a Transaction Reply

To allow you to process new Transaction Reply Functions in Transaction Reply messages from Central, you need to author the processing of the function in the work group of a Tr Reply Function worker. Additionally, you need to specify the attributes for the Tr Reply Function worker as follows:

- Tr Reply Function ID - set this attribute to 'User Function ID' when you are processing a new Transaction Reply Function
- User Function ID - specify a unique Function ID for the new function. The ID must be a single character
- Execute And Wait - specify what the Transaction Handler should do on receiving a Transaction Reply message. The possible values for this attribute are:
  - True - executes the function and then waits in the Transaction Handler for another Transaction Reply message
  - False - terminates the transaction.

When the Transaction Handler attempts to process the Function ID, it checks its 'Tr Reply Function' work group to verify whether a worker for this Function ID exists. If it does exist, the user function is processed.

An example of when you may want to process a new Transaction Reply Function is to support a third party device.

If, during a user function for a new device, a solicited / unsolicited status message is required for the device, you need to build and send the status information. Typically the status information would be retrieved using the ActiveX control for the new device.

If a user function sends a solicited status message to Central, then it must set the Solicited Status Message Sent Flag CDI store. This is necessary so as to inform the Transaction Handler that a solicited status message has been sent. The Solicited Status Message Sent Flag CDI Store is described in Appendix C, "Common Data Interface Stores".

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

New Printer Flags in a Transaction Reply

To allow you to add new printer flags to a Transaction Reply, any unrecognised printer flags and their associated data are placed in CDI stores named:

PrinterFlag<ASCII character used for printer flag>

For example, if '?' was used as the printer flag then the name of the store would be PrinterFlag?.

When you want to print the data, you would access the appropriate CDI store and print the data to the printer required.

NDC Transaction Handler

You only need to use the NDC Transaction Handler worker if you have replaced the NDC+ Transaction Request State with worker hierarchies.

The NDC Transaction Handler has an 'NDC Field' work group that is used to specify which optional component fields are included in a Transaction Request message. When you use this worker, it is your responsibility to populate the 'NDC Field' work groups with the CDI stores used in the Transaction Request message (for example, Amount Buffer, Buffer A, Operation Code Buffer and Card Track 1 Data). All mandatory fields and Field / Group separators are automatically included in the message.

We have provided a copy of the NDC Transaction Handler worker, in the 'NDC Transactions' catalog, which contains workers to execute all NDC Transaction Reply functions. All you need to do is copy this worker and specify which optional Transaction Request fields you wish to include.

The following section details the relationship between the NDC Field workers and the CDI stores.

For details of CDI stores, see Appendix C, "Common Data Interface Stores".

NDC Fields and CDI Stores

NDC Field Workers are provided in the 'NDC Field Workers' catalog for each of the optional fields that can be included in the Transaction Request message.

To include non-mandatory fields in the NDC Transaction Handler, you need to place the relevant NDC Field workers into the 'NDC Field' work group, in any order. Before activating the NDC Transaction Handler, you should ensure that you have written to the corresponding CDI store workers.

The NDC Transaction Handler will automatically ensure that all mandatory fields are included, along with any Data Identifiers, Field Separators and Group Separators. It will also include data for

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

the optional fields you have selected through using the NDC Field workers.

The following table shows the relationship between NDC Field workers and CDI stores by providing the following information:

- Field - Field in the Transaction Request message
- M / O - Whether the field is Mandatory / Optional
- NDC Field Worker - Name of the NDC Field worker
- Common Data Store - CDI store where data for the field is taken from
- Catalog - Catalog where the CDI store worker is contained
- Field Description - Use of each field, and any other CDI stores that influence whether a field is actually included or not.

Table 7-1
NDC Field Workers and CDI Stores

Field	M/O	NDC Field Worker	Common Data Store	Catalog	Field Description
b	M	N / A			Message Class. Value of '1'.
c	M	N / A			Message Subclass. Value of '1'.
d	M	N / A	Logical Unit Number	CDI - Terminal Configuration	Logical Unit Number.
			MAC Machine Number	CDI - Security	If the "MAC Machine Number" is not six spaces, the data from this Common String Store is included with the Logical Unit Number as part of field "d".
e	O	Time Variant No.			Time Variant Number. If this NDC Field is specified, the Time Variant is calculated at the time of sending the Transaction Request, rather than taking data from a Common Data Store.
f	M	N / A	Top of Receipt Flag	CDI - Transaction Processing Flags	Top of Receipt Flag.
g	M	N / A			Message Coordination Number. This field is calculated at the time of sending the Transaction Request, rather than taking data from a Common Data Store.
h	O	Track 2	Card Track 2 Data	CDI - Buffers	
i	O	Track 3	Card Track 3 Data	CDI - Buffers	

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Field	M/O	NDC Field Worker	Common Data Store	Catalog	Field Description
j	O	Op Code Data	Operation Code Buffer	CDI - Buffers	
k	O	Amt. Entry Field	Amount Buffer	CDI - Buffers	
l	O	Pin Buffer (A)	Buffer A	CDI - Buffers	General Purpose Buffer A. If requested for inclusion, this Common String Store should contain a 16 character PIN, encrypted as specified in the FIT table. The PIN Entry State writes to this buffer. **If the PIN Entry State is not used, it is your responsibility to write to this buffer.**
m	O	GP Buffer (B)	Buffer B	CDI - Buffers	
n	O	GP Buffer (C)	Buffer C	CDI - Buffers	
o	M	N / A			Data Identifier. This field will be a '1' if Track 1 Data is requested for inclusion in the message.
p	O	Track 1			
q	M	N / A			Data Identifier. This field will be a '2' if Transaction Status Data is requested for inclusion in the message.
r	O	Transaction Status	TSN	CDI - Security	Last Transaction Status. If requested for inclusion, this field is made up of the data taken from all of the Common Data Stores indicated.
			Last Message Status	CDI - Security	
			Notes Last Dispensed Cassette 1 Count	CDI - Note Counters	
			Notes Last Dispensed Cassette 2 Count	CDI - Note Counters	
			Notes Last Dispensed Cassette 3 Count	CDI - Note Counters	

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Field	M/O	NDC Field Worker	Common Data Store	Catalog	Field Description
			Notes Last Dispensed Cassette 4 Count	CDI - Note Counters	When option 76 is set to the default (000), the original format is used for Note Counts. This includes the counts for cassettes 1 to 4 only. However, when option 76 is set to support up to seven cassettes (001), the counts for cassette types 5, 6 and 7 are also included.
			Notes Last Dispensed Cassette 5 Count	CDI - Note Counters	
			Notes Last Dispensed Cassette 6 Count	CDI - Note Counters	
			Notes Last Dispensed Cassette 7 Count	CDI - Note Counters	
			Coins Last Dispensed Hopper 1 Count	CDI - Coin counters	When Option 79 is set to the default (0), supporting four hopper types, this field reports the number of coins dispensed from each hopper type. However, when option 79 is set to support more than four hopper types (1), this field contains zeros and the coin counts are reported using buffer 'f'.
			Coins Last Dispensed Hopper 2 Count	CDI - Coin counters	
			Coins Last Dispensed Hopper 3 Count	CDI - Coin counters	
			Coins Last Dispensed Hopper 4 Count	CDI - Coin counters	
av	O	CSP	New PIN Buffer 1 Present Flag	CDI - Buffers	CSP Data. If requested for inclusion, the Data Identifier of "U" is included. If the "New PIN Buffer 1 Present Flag" is non-zero, the data from "Customer Selectable PIN Buffer 1" is also included in the message. The "Customer Selectable PIN Buffer 1" should contain a 16 character encrypted PIN.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Field	M/O	NDC Field Worker	Common Data Store	Catalog	Field Description
			Customer Selectable PIN Buffer 1	CDI - Buffers	
aw	O	Confirmation CSP	New PIN Buffer 2 Present Flag	CDI - Buffers	CSP Confirmation Data. If requested for inclusion, the Data Identifier of "V" is included. If the "New PIN Buffer 2 Present Flag" is non-zero, the data from "Customer Selectable PIN Buffer 2" is also included in the message. The "Customer Selectable PIN Buffer 2" should contain a 16 character encrypted PIN.
			Customer Selectable PIN Buffer 2	CDI - Buffers	
ca	O	N / A	Accept Count1 to Accept Count50	CDI - BNA Accepted Denomination Counts	Bunch Note Acceptor (BNA). If requested for inclusion, this field is made up of the data taken from the Common Data Stores indicated.
			Note1 Active to Note50 Active	CDI - BNA Active Banknotes	
			DenomConfig Type 1 to DenomConfig Type 50	CDI - BNA Denomination Configuration	
			Denomination Type 1 to Denomination Type 50	CDI - BNA Deposited Denomination Counts	
			Denomination Type 1 to Denomination Type 50	CDI - BNA Retract Counts	
			Miscellaneous	CDI - BNA MISC	

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Field	M/O	NDC Field Worker	Common Data Store	Catalog	Field Description
cf	O	Buffer ‘f’	Buffer ‘f’	CDI - Buffers	Last Transaction Status if more than 4 coin hopper types are present. If requested for inclusion, field ‘ce’ is used instead of field ‘r’. Included only if the last Transaction Reply requested a coin dispense and more than four hopper types are present. This field is repeated for each hopper type specified within the Transaction Reply.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Extending the Runtime

If you need to perform a task which cannot be implemented using the Workers or NDC+ State Types we provide, there are other programming tasks you can perform. The following table identifies the various programming tasks available for extending the Advance NDC runtime.

Table 7-2
Extending the Advance NDC Runtime

Advance NDC / NDC+	Programming Task	Refer to ...
Advance NDC	Create an ActiveX Control	*APTRA Author User's Guide* (b66038.pdf)
	Write an external C function	*APTRA Advance ADE, Programmer's Guide* (b66042.pdf)
	Create a new C++ Worker Class	
NDC+	Configure new NDC+ States and Screens	*APTRA Advance NDC, Reference Manual* (b66180.pdf)
	Create new NDC Exits	*Using NDC Exits* (b65102.pdf) CEN-XFS Exits in Advance NDC (white paper)
	Migrate existing NDC+ Exits	"Migrating Existing NDC+ Exits" on page 3-10

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Chapter 8

Enhancing the Application Core or Supervisor

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and/or distribution is strictly prohibited.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Overview

In NDC+, you could customise your self-service application using States / Screens. With Advance NDC, customisation can also be performed by modifying the Customisation Layer, Application Core or Supervisor.

This chapter describes how to modify the Application Core if you want to do any of the following:

- Process new Message Classes
- Include additional data in Terminal State messages
- Process new Enhanced Configuration Parameters
- Add specific routines to the Application Core, such as the following:
  - — Initialise Start Up
  - — Override configuration options / terminal commands
  - — Add a new Supervisor menu / function authored in the APTRA Author

  Note 1: If you plan to customise the Application Core, take into consideration the following:

  - Care must be taken to leave the core NDC+ functionality (mode handling and existing messages) intact
  - Any customisations you make will have to be re-applied to future service packs for and new releases of Advance NDC.

  Note 2: The Supervisor functionality is in a separate Supervisor application. For details, see “Supervisor Mode” on page 6-26.

  Note 3: Functionality for User Messages and User Terminal Data is in the Application Core, not in separate ActiveX controls.

If you intend modifying the Customisation Layer, see Chapter 7, “Enhancing the Customisation Layer”.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Customising the Application Core

The following table identifies parts of the Application Core you can customise, and references sections for details of how to make the changes.

Note: To accommodate some of these changes, the message interface must be extended, requiring a change at Central.

Table 8-1
Customisable Parts of the Application Core

Customisation	How?
Process a new Message Class or include additional data in a Terminal State message	Modify the authored flow for the Application Core author project for each mode as required. *See* the following sections: "Overview of User Messages / Terminal Data" on page 8-2, "Before Modifying the Application Core" on page 8-4, "Processing a New Message Class" on page 8-5, "Adding Additional Data to Terminal State Messages" on page 8-11.
Processing new Enhanced Configuration Parameters	Create and use User-defined Common Data Interface (UCDI) stores. *See* "Processing new Enhanced Configuration Parameters" on page 8-15.
Add specific routines to the Application Core (for example, to initialise Start Up, override configuration options / terminal commands, or add a new Supervisor menu / function authored in the APTRA Author)	Modify the authored flow for the Application Core author project. This needs to be done with care. For an example of adding a new Exit Supervisor menu and authored Supervisor menu, see "Specification of Exit Supervisor Menus" on page 3-18 and "Supervisor Mode" on page 8-18.

Overview of User Messages / Terminal Data

The Message Handler, described in "Message Handling" on page 6-19, processes the message class / terminal command. If the command code is outside the expected range, the Message Handler sends an Unknown Message or Add Terminal State response to activate the User Messages or User Terminal Data work group. The User Messages and User Terminal Data directors handle messages in a similar way.

Care needs to be taken to ensure you use the correct values for the signals. The signals and their values are specified in "User Stores and Signals Catalog" on page 6-15.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Note: When browsing the authored examples for User Messages and User Terminal Data, notice how these signals are used. NCR recommends you browse the authored examples in conjunction with reading “Processing a New Message Class” on page 8-5 and “Adding Additional Data to Terminal State Messages” on page 8-11.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Before Modifying the Application Core

This section provides important information that you should be aware of before modifying the Application Core.

Compatibility Considerations

If you intend extending the NDC Message Interface to process a new Message Class or add additional data to Terminal State messages, your extensions must remain compatible with Advance NDC. Therefore, you must contact NCR Dundee Product Management to obtain unique message identifiers. Examples of when you need to obtain message identifiers are for a new:

- Message Class
- Command Code
- Field in a message (such as a new data field in a Transaction Request)
- Data Identifier in a message
- Function ID in a Transaction Reply
- Printer Flag in a Transaction Reply
- DIG (Device Identifier Graphic) for a new device.

Preparation Guidelines

You must follow the general guidelines “Preparation Guidelines” on page 7-8 when preparing to modify the Application Core, as the guidelines apply to any application before modification.

Before and during these guidelines, it is good practice to back up the provided projects, and the projects you are working on.

Customisation Guidelines

When you make changes to the Application Core, you must save the Application Core Author project (*ApplicationCore.mpj*) with a different name and back it up, so as to avoid overwriting your changes when re-installing Advance NDC.

Remember: if you extend the NDC Message Interface, you need to contact NCR to obtain unique message identifiers.

If you make changes to the Application Core, these changes will have to be re-applied when moving to subsequent releases of Advance NDC, or installing service packs for Advance NDC. You also run the risk of breaching the SST to Central message interface.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Processing a New Message Class

The User Messages director lets you add a new message class or terminal command to the Application Core authored flow for each mode. The User Messages director is supplied in the *ApplicationCore.mpj* Author project. To find out where the User Messages work group is used in the application flow, see "Message Handling" on page 6-19.

The following sections describe the default implementation of the User Messages director in the Application Core, a summary of the procedure for authoring a new Message Class, and an example implementation of processing a new Message Class.

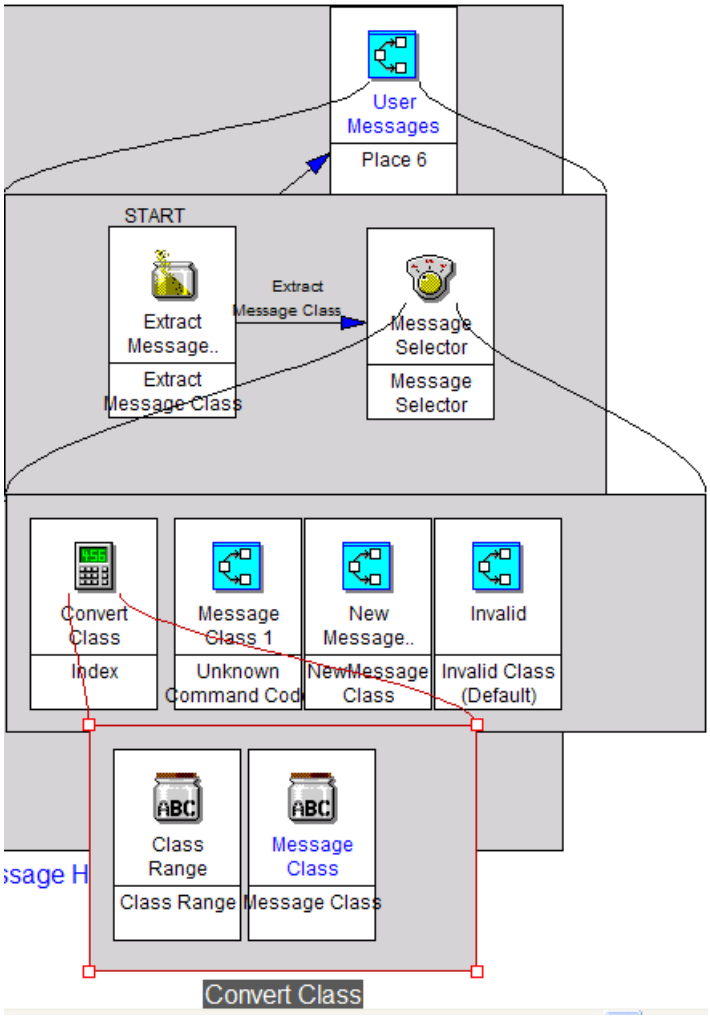
Default User Messages Implementation

The User Messages authored flow extracts the message class from the received message and processes the message accordingly. The default implementation is to reject an unknown message class or terminal command.

The following diagram shows a view of the User Messages Director.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Figure 8-1
User Messages Director



To open a view of the default authored User Messages for the Out of Service mode, do the following:

1 Open the Application Core Applications catalog and double-click the Application Core application.

2 Navigate to the OOS Message Handler (component ID 11l286b7)

3 Double-click the User Messages Director.

Selector workers are used to decompose a message received from Central. The general principle for decomposing a message is to first look at the Message Class, followed by the Command Code, followed by the Modifier.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Extract Message Class

Extracts the message class (the first byte of data) from the Central message.

Message Selector

Activates the appropriate work group depending on whether a new message class or a terminal command with a new command code is to be processed.

Note: When a Selector is activated, it uses its Index work group to decide which Doer work group to activate. For more information, refer to the on-line help.

The work groups are as follows:

- Convert Class - Index to check whether a new message class or a terminal command with a new command code is to be processed. It does this by looking at the Class Range store and comparing this with the contents of the Message Class store.

  Note: To ensure that terminal commands with invalid command codes are rejected, the value '1' (representing the message class for a Terminal Command) must always be included in the Class Range store as the first value.

- Unknown Command Code - by default, this Director rejects a terminal command with an unknown command code.
- New Message Class - this Director is a placeholder for you to author how to process a new message class.
- Invalid Class - if the message class is not recognised, this Director sets up the reject status and returns a reject signal. You should not modify this functionality.

For an example implementation of the User Messages director, *see* "Example User Messages Implementation" on page 8-8.

Summary of Procedure

To process a new message class / terminal command, complete the following:

1 Obtain the required Message Class number from NCR Dundee.

2 Append the class number to the Class Range store.

3 Author the relevant Director (New Message Class | Unknown Command Code) to process the message for the required mode by adding work groups and workers as required.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

4 Include the Invalid Cmd Code work group, which is activated if the command code is not recognised.

5 Whenever you want to send a reject signal, set the Reject Status Value and Qualifier CDI store to the appropriate reject value (for example, 'C02').

Note 1: If the message has to be processed differently in each mode (for example, processed in In Service, rejected in Out Of Service, and held for processing later in Supervisor), then the Current Mode CDI store has to be checked and the relevant processing executed for each mode.

Note 2: If you are processing an OOS mode message, put the message in the Message For Host store. The format of this message is entirely up to you provided Central can deal with the format.

Example User Messages Implementation

NCR provides an example implementation of the User Messages director to show how you could process a new message class from Central to the Terminal.

To open a view of the example implementation of the authored User Messages, do the following:

1 Open the User Control Examples catalog.

2 Double-click the User Messages Example.

3 Navigate down the worker hierarchy to view the Message Selector worker.

Message Selector

As in the default implementation of the authored User Messages, the Message Selector activates the appropriate work group depending on whether a new message class or a terminal command with a new command code is to be processed.

Here, the new message class is '9' and the value of the Class Range store is '19', indicating the following:

- Terminal commands '1', are processed
- A new message class of '9' is processed using the second Doer work group.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Unknown Command Code Director

Processes a new terminal command code. It contains two work groups:

- Deformat Message - breaks up the Terminal Command message received from Central into its constituent parts for further validation and processing. The significant fields are the Command Code and Command Modifier
- Process Message - processes the Terminal Command based on its command code. This is discussed in the following section.

Process Message Contains three work groups:

- Index - checks if a new terminal command code is to be processed. It does this by looking at the Cmd Code Range store and comparing this with the contents of the Command Code store. In this example, the Cmd Code Range store indicates that command codes of type 'a' are supported.
- Process a - processes a command code of type 'a' based on the current mode:
  - Out Of Service - fills the Status Info store with the status information and generates a send terminal command signal
  - Supervisor - stores the terminal command for processing in the next mode (that is, Out Of Service or In Service). The terminal command is stored as follows:
    - — 'Pending TC' UCDI store is initialised
    - — Contents of the 'Host Message' UCDI store is copied to the 'Pending TC' store
    - — 'Pending TC Msg Flag' is set to True to indicate there is a pending terminal command
    - — 'Stop Suspend' Caller
    - — 'Processing Complete' signals to indicate that no reply is to be sent to Central
  - In Service - stores the terminal command if a transaction is in progress; otherwise the command is rejected
- Invalid Cmd Code - if the command code is not recognised, this work group is activated. It sets the reject code to 'B04' (Invalid Command Code) and then returns a reject signal.

New Message Class

Processes messages with a Message Class of '9'. It contains two work groups:

- Copy Mode to String - uses an Assigner to convert the value of the Current Mode CDI store from an integer to a local string store named Copy of Mode

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

- Mode Selector - processes the Message Class based on the current mode. The Convert Mode Integer Computer worker uses the Copy of Mode local string store to select the 'mode' work group to activate:
  - — Out Of Service: sends a new message to Central in reply to the new message class.
  - — Supervisor: sends a Ready9 response to the new message class
  - — In Service - sends a reject response to the new message class.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Adding Additional Data to Terminal State Messages

The User Terminal Data director lets you add additional terminal status data to the following Terminal State Solicited Status Messages:

- Supply Counters
- Tally Information (*See* Note:)
- Error Log Information (*See* Note:)
- Hardware Configuration Data
- Supplies Data
- Fitness Data.

Note: In Advance NDC 3.0, this returns a preconfigured default date and zero values.

The User Terminal Data director is used in the Application Core's authored flow for each mode. The User Terminal Data director is supplied in the *ApplicationCore.mpj* Author project. To find out where the director is used in the application flow, see "Message Handling" on page 6-19.

The following sections describe the default implementation of the User Terminal Data director in the Application Core, a summary of the procedure to add additional data to Terminal State messages and an example implementation.

Default User Terminal Data Implementation

The default implementation of User Terminal Data returns a signal to the script host without adding to the Terminal State information.

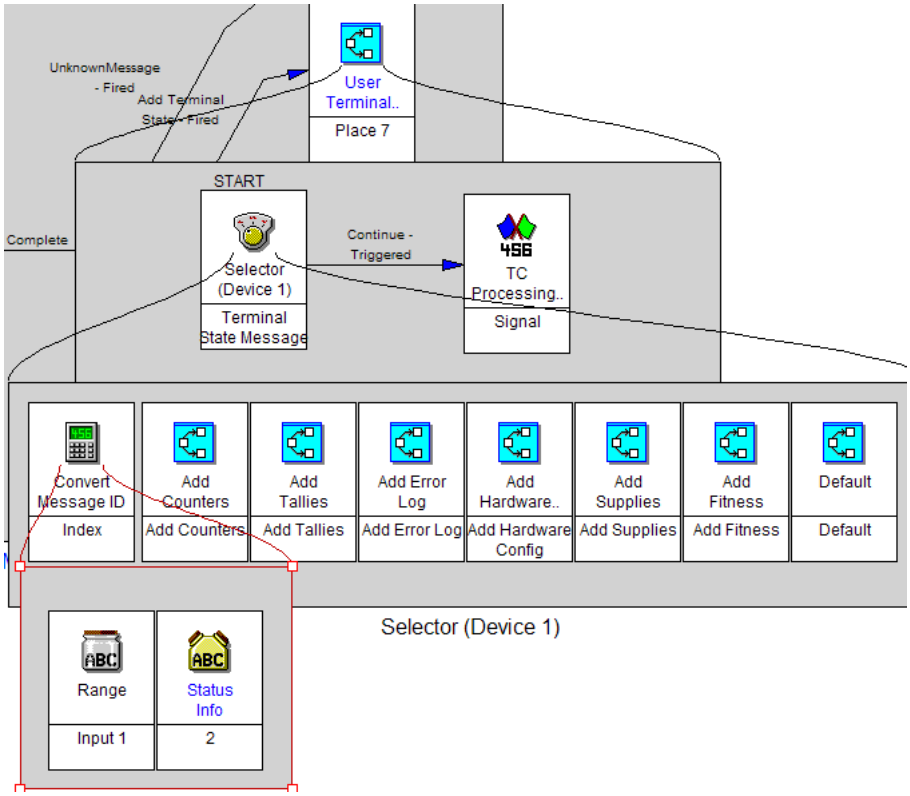
To open a view of the default version of the authored User Terminal Data director for the Out of Service Mode, do the following:

1. Open the Application Core Applications catalog and double-click the Application Core application.
2. Navigate to the Message Handler (component ID 11l286b7).
3. Double-click the User Terminal Data Director.
4. Double-click Selector (Device 1).

The following diagram shows a view of the User Terminal Data Director.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Figure 8-2
User Terminal Data Director



The Status Info CDI store contains the Terminal State information (that is, field 'g' of the Terminal State message). The first byte of this information is the Message ID, which uniquely identifies the type of status information: Counters (2), Tallies (3), Error (4), Hardware (H), Supplies (I) and Fitness (J).

The User Terminal Data authored flow obtains the Message ID by looking at the Status Info CDI store and comparing it with the Range store. It then activates the appropriate work group:

- Add Counters - by default, this Director simply continues without adding any data to the terminal state message
- Add Tallies - by default, this Director rejects an unknown Group Number. You can author the functionality to process a new Group Number, but you must ensure you always reject unknown Group Numbers

  Note: Reporting of tallies is not supported in Advance NDC.

- Add Error Log - by default, this Director simply continues without adding any data to the terminal state message

  Note: Reporting of error logs is not supported in Advance NDC.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

- Add Hardware Config - by default, this Director simply continues without adding any data to the terminal state message
- Add Supplies - by default, this Director simply continues without adding any data to the terminal state message
- Add Fitness - by default, this Director simply continues without adding any data to the terminal state message
- Default - this Director should never be needed.

Procedure for Adding User Terminal Data

If you want to add additional data to one or several of the Terminal State messages, you would author the appropriate Director to add the information to the Status Info store. For example, to add Fitness Information for a new device, you would add a group separator and then the Device ID and fitness data for the device to the end of the Status Info store.

To add additional data to a Terminal State message, complete the following:

1 Obtain a new identifier from NCR Dundee for any new device.

2 For the required mode, navigate down the worker hierarchy to show the Selector (Device 1).

3 Author the relevant director (Add Counters, Add Hardware Config, Add Supplies and/or Add Fitness) by adding work groups and workers as required.

   For example, to add Fitness data for a new device add a group separator, device ID and the fitness data for the device to the Fitness director. This is automatically added to the end of the Status Info store.

4 If you have more than one new device, author a different Selector to add terminal state data to each message.

For an example implementation of the User Terminal Data director, see the next section.

Example User Terminal Data Director

The example implementation of the User Terminal Data director shows additional data to all Terminal State messages.

To open a view of the example implementation of the authored User Terminal Data director, do the following:

1 Open the User Control Examples catalog.

2 Double-click the User Terminal Data Example application.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and/or distribution is strictly prohibited.

3 Navigate down the worker hierarchy to show the Selector (Device 1).

Selector (Device 1)

Like the default version of the User Terminal Data director, the Selector (Device 1) obtains the Message ID by looking at the Status Info CDI store and comparing it with the Range store. It then activates the appropriate work group:

- Add Counters - appends additional Counter information to the terminal state message
- Add Tallies - not supported
- Add Error Log - not supported
- Add Hardware Config - appends additional hardware configuration information to the hardware config terminal state message
- Add Supplies - appends additional supplies information to the supplies terminal state message
- Add Fitness - appends additional fitness information to the fitness terminal state message.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Processing new Enhanced Configuration Parameters

The Message Handler performs the standard processing for the Enhanced Configuration Parameters Load message. All the standard timers and options are processed.

If there are any new timers or options (not supported by the standard NDC+ product) in the message, these will be put into UCDI stores named Timer<Timer No> and Option<Option No> respectively.

You can access these stores via the Author by creating a CDI store with its 'Store ID' attribute set to zero and its 'Store Name' set to Timer<Timer No>/Option<Option No>.

For information on UCDI stores, see Chapter 9, "Using User-defined CDI Stores".

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and/or distribution is strictly prohibited.

Altering Modes

As stated in “Mode Handling” on page 6-17, you should not change the Mode Handler functionality. To prevent you from accidentally modifying the Mode Handler functionality, its workers have been assigned to a different module from those parts of the application you can modify.

It is possible to add functionality to each of the Modes contained within the Mode Handler, but the utmost care needs to be taken when making any changes.

For example, if a mode change is called while a concurrent operation is running, the operation may not finish, thus leaving the operation in an incomplete state. This could cause problems if the application relies on the completion of this operation. It is therefore important that changes introduced to a mode are done so in an appropriate manner. This could mean inserting an operation sequentially (with little impact) or adding a concurrent operation that requires the overhead of synchronising with other concurrent operations.

Note: Any changes made to a mode will need to be re-applied when re-installing Advance NDC, or installing a subsequent release or service pack.

Each of the modes follows roughly the same pattern:

- Initialisation
- Processing
- Mode change.

The easiest parts to modify, with the least impact, are at entry to and at exit from a mode. This is because several concurrent activities may be taking place during processing, and some method of synchronisation is required, increasing the complexity of the application. It is therefore recommended that changes should be made on entry to and exit from a mode.

Start of Day and Initialise Tasks

Any new activities added to the Start of Day or Initialise tasks must synchronise their activities with the current task. Activities can either be inserted in the flow of the initialisation and executed sequentially, or added concurrently, synchronising completion with the main flow of the task. Inserting an activity into the flow is the easiest to do, having the least impact on the mode.

You should determine the correct task to alter. Note that the Initialise task checks the position of the mode switch before

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

progressing; also note that there is a potential for the Start of Day and Initialise tasks to enter straight into a mode in the main Mode Handler, based on the configuration and previous state of the SST.

In Service Mode

In Service mode synchronises activities with the Customisation Layer. You should have a clear understanding of these interactions and how any changes they make may affect this synchronisation.

There are five distinct parts to In Service mode:

1 **Entry to the mode** - performs any initialisation and processing of queued messages.

2 **Initiating Customisation Layer synchronisation** - once the Customisation Layer has started the SST can be thought of as being 'In-Service'.

3 **Processing Terminal to Central Messages** - messages are handled differently based on the message class and the status of the Customisation Layer. Messages are generally processed while the Customisation Layer is 'IDLE' or 'WAITING'.

4 **Suspended** - if Customer Tampering is detected, the Customisation Layer will be deactivated and Suspend will be entered.

5 **Leaving the mode** - a mode change has occurred and the current mode is being left. The Customisation Layer must be stopped before the mode change can progress.

Changes can be made at any of these points with the most impact occurring at points 3 and 4. Messages should be processed as quickly as possible and holding up this process may affect the running of the Customisation Layer. In Service mode is more complicated than the others, as there are a lot of synchronisation issues. NCR recommends that you add any In Service changes to the Customisation Layer, as there is an already established mechanism for the processing of a Cardholder Session.

Out Of Service Mode

The main activity in this mode is message processing. Configuration Data Load and Terminal Commands can be processed, as well as 'Print Immediate' Transaction Replies. Since you are protected from altering the processing of these messages, any changes you make to this mode should not interfere with these operations.

Offline Mode

Offline mode simply waits for communications to be restored or Supervisor mode to be entered. As the restoration of communications may occur at any time, any actions performed in

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

this mode should be synchronised with the restoration of Communications.

Supervisor Mode

Supervisor mode is your means of controlling the terminal by giving you access to a range of menus and options.

You can customise the Supervisor mode supplied with Advance NDC either with or without the use of the APTRA Author. Additional features are available when the Author is used. You can incorporate additional menus (defined by user exit code or authored additions), add new functions to existing menus, or re-implement functions provided.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Enhancing the Supervisor Application

You can modify the provided Supervisor application, or even replace it completely by re-authoring the application. However, NCR recommends that where possible you customise Advance NDC without using APTRA Author.

If using the Author is necessary, limit changes to those described below. Other changes carry a risk of breaching the existing message interface and mode transition protocol with the Application Core.

Customisation without Using APTRA Author

Text used for displaying menus and messages (acknowledgement, error, prompt, status and information) is contained in reserved screen groups A, E/e, I/i, M/m, P/p and S/s. You can modify this text by editing the screen definitions in the reserved screen file, *resrvd.def* (make a backup copy first), or by downloading new definitions for the relevant reserved screens. Similarly, it is possible to modify the text used for printing Security Trace Messages (screen group T). In this way, you can localise text using the same methods as with NDC+, without needing to use the APTRA Author.

Customisation Using APTRA Author

If you plan to customise the Supervisor application, take into consideration the following:

- Any customisations you make will have to be re-applied to future service packs and updates for Advance NDC
- To maintain compatibility with NDC+, NCR recommends that you do not modify, enhance or re-implement *existing* menu functions. In particular, you should avoid making changes to the Select Menu Exit function, which interfaces with the Application Core when causing a transition from Supervisor mode to another mode.

Text from the reserved screen groups A, E/e, I/i, M/m, P/p and S/s can be replaced by removing the Reserved Screen Retriever workers and redefining the text in the authored application.

You can add new functions to the menus provided, by adding a work group to the relevant Selector and placing a worker providing the new function (typically a Director) in the appropriate work group. The new function is normally placed in the penultimate work group in the Selector, as the last one provides the default functionality. When adding new functions, you should amend the menu display text accordingly.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and/or distribution is strictly prohibited.

The menu function value is set before an Unsolicited Supervisor Keys Status Message is sent to Central. If you wish to change the value reported, you will have to set the new value explicitly.

You can define new menus in the same way as adding new functions to existing menus. In turn, the new menus may have new functions added to them.

As with adding a new Message Class or Terminal Command, you should ensure that signals are correctly consumed by sharing the relevant Pass-thru worker, such as 'Task Complete'.

If you use Operator Display workers and a Supervisor Data Collector in the same work group, the Supervisor Data Collector *must* be placed as the last worker to be activated. If the Supervisor Data Collector is not the last worker, it blocks all Operator Display operations that are activated later until the Supervisor Data Collection is complete. This occurs because the Service Provider processes `WFS_CMD_TTU_READ` and `WFS_CMD_TTU_WRITE` commands sequentially.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Chapter 9

Using User-defined CDI Stores

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Overview

As discussed in Chapter 1, “Introducing Advance NDC” under the heading “User-defined Common Data Interface” on page 1-12, you can create User-defined Common Data Interface (UCDI) stores to share data between the Customisation Layer and Application Core.

In the following sections we discuss:

- User-defined Common Data Interface (UCDI) service
- UCDI initialisation file
- Persistence levels of CDI stores
- Creating a UCDI store.

You must ensure all UCDI stores are cleared, initialised and set by your application, as required.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

UCDI Service

The UCDI is a Windows XP service that automatically runs at system start-up, and implements new CDI stores - which we refer to as User-defined CDI (UCDI) stores.

The UCDI service loads the UCDI initialisation file which initialises the stores declared in the file. For details of the UCDI initialisation file, see “UCDI Initialisation File” on page 9-3.

The UCDI service contains an ActiveX class called ‘APTRAUCDI’, which exposes a COM interface called ‘APTRAUSERCDI’. The class enables access to UCDI data via an Active Script Host worker.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

UCDI Initialisation File

The UCDI initialisation file is a text file named *UCDIini.txt*. It contains details of all known UCDI stores and is loaded at system start-up.

All UCDI stores will be declared in this file as soon as they are created (see "Creating a UCDI Store" on page 9-5). The name, data value and persistence indicator of the store is held in the file. The required format for the file and an example of the file is shown below.

Table 9-1
UCDI Initialisation File Format

Format of Initialisation File	Example of Initialisation File
Name | Data | Persistence_Indicator	BnaNoteCount1 | 32 | 1
Name | Data | Persistence_Indicator	BnaLanguage | English | 3
…	SystemStartTime | 07:02:12 | 1
…	EmptyString | | 1
Name | Data | Persistence_Indicator	

Note: Any lines in the file which do not conform to the required format are ignored.

As the file is a standard ASCII text file, you can edit it with a text editor thus enabling easy configuration of systems.

The 'Name' element represents the name of the store. Each name element causes a new store with that name to be created. Any name conflicts (where the name appears more than once) are resolved on a first come first served basis.

The 'Persistence_Indicator' identifies the level of persistence of the store. It can be 1 (Non-persistent), 2 (Persistent and cannot be reset) or 3 (Persistent and can be reset). See "Persistence Levels" on page 9-4 for more details.

Note: Care must be taken when editing this file for persistent stores. If you edit the data value for an existing persistent store (that is, the store exists in persistent memory already), the value will not be assigned to the store.

String stores can be initialised to contain an empty string. For example, `BnaLanguage||3`.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Persistence Levels

Persistence of UCDI stores is maintained using a binary file named *UCDIpers.dat*.

The UCDI initialisation file contains details of all known UCDI stores, including the names and initial values of persistent UCDI stores.

Table 9-2
Persistence of UCDI Stores

Level of Persistence		Behaviour
1	Non-persistent	The store will maintain the last assigned value as long as the UCDI service is running. If the UCDI service is restarted, the store will contain the value defined by the initialisation file.
2	Persistent and cannot be reset	The store will maintain the last assigned value throughout system shutdown and restart. The store cannot be reset to its initial value by SST configuration changes.
3	Persistent and can be reset	The store will maintain the last assigned value throughout system shutdown and restart. The store may be reset to its initial value by a call to the `ResetPersistentStores` method.

If you need to reformat the terminal's hard disk, first back up *UCDIpers.dat*.

If you want to clear all persistent data, you can delete *UCDIpers.dat*—it will be recreated on system start-up. The persistent stores will take on the values defined in the initialisation file.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Creating a UCDI Store

UCDI stores must be created before they can be accessed. In the following sections we discuss the three methods available for creating a UCDI store:

- Method 1 - Using the UCDI initialisation file and a Common Store
- Method 2 - Using an Automation Object
- Method 3 - Using the CreateObject function.

Method 1 - Using the UCDI Initialisation File and Stores

1. Specify the store in the initialisation file. For details of the format of the initialisation file, see “UCDI Initialisation File” on page 9-3.

2. Re-boot the system.

To access the UCDI store, use a Common Integer Store or Common String Store worker with its ‘Store ID’ attribute set to zero and its ‘Store Name’ attribute set to the name of the UCDI store you specified in the initialisation file.

If you attempt to create a UCDI store via a CDI worker without first initialising the store in the initialisation file, then zero will be returned for a Common Integer Store and null for a Common String Store.

To allow for a successful completeness check of applications, the name attribute for CDI store workers is set to an initial value of:

- ‘defaultInteger’ for Common Integer Stores
- ‘defaultString’ for Common String Stores.

Method 2 - Using an Automation Object

1. In an Active Script Host worker, place an Automation Object in its ‘Data Objects’ work group.

2. Set the Automation Object’s ‘Class Identifier’ attribute to `APTRAUSERCDI.APTRAUCDI`.

   This will expose the APTRAUCDI class interface; the methods of which will be shown in the Script Editor, accessed via the Active Script Host. For details of the methods of the APTRAUCDI class, see “APTRAUCDI Properties and Methods” on page 9-6.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

For an example implementation of this method, do the following:

1 Open the *CustomisationLayer.mpj project.*

2 Open the User Defined CDI catalog.

3 Double-click the UCDI test application.

4 Double-click the Active Script Host.

5 View the Automation Object attribute and the VBScript of the Script Host.

Method 3 - Using the CreateObject Function

In an Active Script Host worker, create an instance of the APTRAUCDI class using the CreateObject function.

Once created, the methods of APTRAUCDI class will be available. For details of the methods, see the following section.

The following code sample demonstrates this method.

```
‘variable myCDI declared as global
Dim myUCDI

‘CreateObject used to assign a reference to ‘the
APTRAUCDI class
Set myUCDI = CreateObject("APTRAUSERCDI.APTRAUCDI")

‘can be any function within this Script Host ‘worker
Sub activate()

'data assigned to string store
UCDI.UCDIstring("GreetingMessage") = "Hello, Mr.
Customer"

'data retrieved from store
MsgBox myUCDI.UCDIstring("GreetingMessage") & ",
Welcome from NCR Bank"

MsgBox " Your fast-cash amount is £" &
myUCDI.UCDIinteger("CurrentCustFastCashAmount")

End Sub
```

APTRAUCDI Properties and Methods

The main properties and methods of the APTRAUCDI class are:

- UCDIstring(name) property
- UCDIinteger(name) property

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

- newUCDIstore(name, initval, thetype) method
- ResetPersistentStores() method.

We describe these in the following tables.

Table 9-3
UCDIstring(name) Property

Parameters	name - String representing the Store Name of the target UCDI store.
Assign expression=*x*	The store with Store Name of 'name' is assigned the value of *x*; the value will be treated as a string.
Read *x*=expression	The method will return the value stored in the target UCDI store as a string.

Table 9-4
UCDIinteger(name) Property

Parameters	name - String representing the Store Name of the target UCDI store.
Assign expression=*x*	The store with Store Name of 'name' is assigned the value of *x*; the value will be treated as an integer. Where *x* cannot be converted to an integer, zero will be stored.
Read *x*=expression	The method will return the value stored in the target UCDI store as an integer. If the store contains a string, an integer conversion will be attempted. Zero will be returned if the conversion fails.

Note: This property will cause an Invalid Argument Exception (E_INVALIDARG) if the ‘name’ parameter is not recognised by the UCDI. Use an ON ERROR statement to catch this error value if there is a risk of using unknown names.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Table 9-5
Boolean newUCDIstore(name, initval, thetype) Method

Description	This method creates a new UCDI store with the name and initial value provided. The level of persistence required is indicated by the 'thetype' parameter. This method can also be used to update the default value of a store (but not a persistent store). If the method is called with a name already registered, but a different 'initval' parameter, then the 'initval' value will become the new default value of the store. If the ‘initval’ value is the same as the current default value, the method will return FALSE.
Parameters	name - String representing the Store Name of the target UCDI store. The name must be less than 80 characters and must not contain a '|' (bar) character.
	initVal - string containing the initial value of the new store.
	thetype - string indicating the level of persistence required. The string can be: ● 1 - “Normal” ● 2 - “Persistent” ● 3 - “Reset Persistent”
Return value	A Boolean will be returned. The value of the Boolean will be TRUE if the new store is successfully created. The value FALSE is returned if the store name already exists.

Table 9-6
ResetPersistentStores() Method

Description	This method will set all level 3 persistent stores (that is, stores that can be reset) to their data values defined in the current initialisation file. The operation is reversible if you have made a backup copy of the *UCDIpers.dat* file.
Parameters	None

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Chapter 10

Delivering an Advance NDC Application to the SST

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and/or distribution is strictly prohibited.

Overview

This chapter describes the following:

- Initial installation of the Advance NDC package on an SST, using the Windows XP desktop
- Secure initial unattended installation of the Advance NDC package on an NCR SST
- Configuring the encryptor
- Preparation of modified applications, including migration from NDC+, for installation
- Post-installation activities
- De-installing Advance NDC.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Desktop Installation of the Advance NDC Package

Using the Windows desktop to install Advance NDC on an SST requires you to complete the following:

1. For an installed release of Advance NDC prior to ANDC 3.01 this should be uninstalled before proceeding.
2. Install Advance NDC as provided by NCR.
3. If you have any customisation to apply, install your customisation.
4. Complete the post-installation operations.

Caution: All installations of Advance NDC prior to Version 3.01 and associated components that have been installed separately *must* be removed first. This is done using the **Add or Remove Programs** option in the Windows control panel. In addition, if you separately installed any of the other components supplied with this package, remove them according to the instructions supplied with that component.

Before You Start

The Advance NDC setup file is in the root directory of the APTRA Advance NDC Package CD-ROM, along with the release bulletin and the *APTRA Advance NDC, Overview*. To check you have all the prerequisites to install the software, you can access the overview document as follows:

1. Insert the APTRA Advance NDC Package CD-ROM in the CD-ROM drive of the SST.
2. Locate the file named *APTRA Advance NDC Overview.pdf* in the root directory of the CD-ROM and open it.
3. Ensure the prerequisite non-NCR software is installed on the SST, as described under “Software Requirements” in Chapter 4 of the Advance NDC Overview document.
4. Also check the *Release Bulletin* for the latest additional information.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Installing Advance NDC on an SST

To install Advance NDC, select *setup.exe* from the root directory of the APTRA Advance NDC Package CD-ROM. Follow the on-screen instructions to complete the installation.

SST Directory Structure

The following directory structure is created on the SST when you install the Advance NDC aggregate.

Table 10-1
SST Directory Structure

Directory	Description
c:\SSDS	Advance NDC base directory
c:\SSDS\DLL	Advance NDC component DLL files, user defined DLL files and code page mapping files
c:\SSDS\APPS	Application-related files

The *pmdata* File

From Advance NDC 3.02, deinstalling removes the *pmdata* file.

An empty *pmdata* file is installed as part of the Advance NDC installation. If a *pmdata* file is present on the machine from an earlier installation, the empty *pmdata* file overwrites the existing file.

Note: This does not apply for Service Pack updates.

This approach avoids the need to manually delete the *pmdata* file because of possible incompatible formats across versions of Advance NDC.

Starting the Advance NDC Application

To start the Advance NDC application, select **Start | Programs | NCR APTRA | Advance NDC | Start Advance NDC**.

This activates *startapps.vbs*, which does the following:

1. Checks the platform version.
2. Runs the application components in the following order:
   a. Application Core
   b. Supervisor
   c. Customisation Layer

Between applications, there is a delay defined by the following registry setting:

```
HKLM\SOFTWARE\NCR\Advance NDC\SPStartPause
```

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

The default is 30 seconds. This delay is required to allow the applications to synchronise correctly at start-up.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Secure Initial Unattended Installation (IUI)

This section describes the steps required to prepare Advance NDC for a secure initial unattended installation (IUI) on NCR SSTs running Microsoft Windows XP OEM.

APTRA Advance NDC Updates for APTRA Security

From APTRA Advance NDC 2.6 onwards, all files created at runtime, including the EJ and hardcopy backup files, are located in *C:\Program Files\NCR APTRA\Advance NDC\Data* to allow read/write access under APTRA Security.

The following table gives details of the files that have been moved to *C:\Program Files\NCR APTRA\Advance NDC\Data.*

Table 10-2
Advance NDC File Details

Advance NDC 2.5 Path and Filename	Description
`C:\pmdata\pmpageentry`	Persistent memory file
`C:\windows\system32\custom.dat`	NDC download backup file
`C:\keystore.txt` `C:\keystore.tmp`	Security related files
`C:\FONTDEFS.INI` `C:\FONTDEFS.TXT`	Font related files
`C:\windows\system32\NDCDATAA.DAT` `C:\windows\system32\NDCDATAD.DAT`	System settings backup files

`C:\ssds\apps\ERJCPY.LOG`
`C:\ssds\apps\ERJCPY.IDX`
`C:\ssds\apps\EJDATA.LOG`
`C:\ssds\apps\EJDATA.IDX`
`C:\ssds\apps\HBDATA.LOG`
`C:\ssds\apps\HBDATA.IDX`
	Journal related files

`C:\windows\system32\ucdiini.txt` `C:\windows\system32\ucdipers.dat`	User defined storage files
`C:\ndcrcpt`	*resrvd.def* K07 screen (see Note 1)
`C:\frontimg.bmp`	*resrvd.def* C07 screen

Note 1: This is the path for printer graphic file images. It is not defined in *resrvd.def* for Advance NDC so that printer driver defaults are used.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and/or distribution is strictly prohibited.

Advance NDC Customisations under APTRA Security

NCR recommends that all file references from authored applications are below the *C:\Program Files\NCR APTRA\Advance NDC* directory, under APTRA Security. Any executable files that need to be on the system path should be placed in *C:\ssds\dll*. Any controls or object requiring registration should be placed under *C:\Program Files\NCR APTRA\Advance NDC*.

All installation operations must be carried out under the Administrator account.

Windows XP Security

Deviations from the guidelines laid out by APTRA Security (XP) require modification to the Windows XP local security policy. For information on security guidelines refer to the APTRA Security on-line help under **Security Setting | Local Policies**. You can modify the Windows XP local security policy from the desktop or an installation time script. Modifications other than those laid out by APTRA Security are beyond the scope of this document. For details of modifying the local security policy, refer to the *Microsoft MSDN Library*.

Implementing APTRA Security XP with Advance NDC

Advance NDC is supported by the release of APTRA Security as supplied on the Advance NDC CD-ROM. APTRA Security is supplied as a separate component on the Advance NDC Package CD-ROM. To apply it to an SST, it must be added to your super-aggregate, as described in “Prepare the Advance NDC Super-Aggregate.” on page 10-7.

APTRA Advance NDC and APTRA Initial Unattended Installation

Advance NDC can be installed using the APTRA IUI process. The following topics outline the steps required to prepare the Advance NDC IUI installation. Advance NDC requires two CD-ROMs due to the size of the Windows XP and APTRA XFS installations.

Note: For a summary of the steps involved in creating an Advance NDC IUI, see “Steps in the IUI Process.” on page 10-10. For more detailed instructions on preparing an IUI CD, refer to the APTRA on-line documentation under **Initial Unattended Installation**.

Advance NDC IUI Preparation

Preparing Advance NDC for unattended installation on NCR SSTs requires the following steps:

1. Prepare the Advance NDC super-aggregate.
2. Prepare the CD layout.
3. Update the IUI and Security batch files.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

4 Burn the CD.

Prepare the Advance NDC Super-Aggregate.

A super-aggregate is one aggregate containing all the aggregates and components you intend to install, such as platform, application, service packs and optional components.

Complete the following steps:

1 In the Aggregate Builder tool, create a new aggregate, for example “APTRA Advance NDC 3.02.x”.

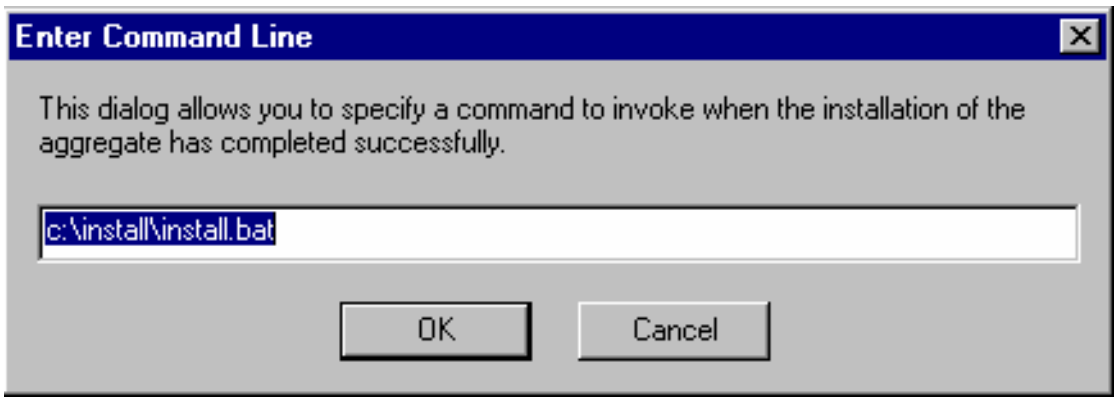
2 Import the APTRA XFS and Advance NDC aggregates into the new aggregate.

3 If you require a secure installation on Windows XP, import the APTRA Security aggregate into the new aggregate.

4 If you want to include an APTRA Advance NDC Package Service Pack, import the Service Pack into the new aggregate.

5 In the properties of the new aggregate, set the **Command** property to *C:\install\install.bat*.

Figure 10-1
Aggregate Properties Dialog



6 Export the new super-aggregate to the EXTEND directory, which will be placed on the second CD-ROM layout.

Note: As the aggregate contains APTRA XFS, it must be installed by a command within *cmdlines.txt*.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Prepare the CD Layout.

IUI (XP) installs a skeleton deployment archive that contains the template files described in this chapter. This deployment archive may be used as a starting point when preparing CD-ROM files.

Prepare the CD layout on your development PC as follows:

1. Copy the IUI directory structure to your working area. This will be CD1 (called *SSTCD* in the IUI documentation).

2. Copy the Windows installation files to CD1 as described in the APTRA IUI documentation.

3. Copy the *EXTEND* directory containing the super-aggregate to the CD2 layout.

The folders in *SSTCD* (CD1) after you have prepared the files are as follows:

- *INSTALL* - Information common to all SSTs
- *UNIQUE* - Information unique to each individual SST

Update the IUI and Security Batch Files.

The following files need to be modified:

- *extendcp.bat*
- *install.bat*
- APTRA Security batch files

Exrtendcp.bat Update this file to reflect your IUI. Change the references to *ModelDir* to *ANDC* and *ModelFil.txt* to *ANDC.txt*.

Place a text file named *ANDC.txt* into the *EXTEND* folder on CD2. This empty marker file indicates that the folder contents are to be copied to the hard drive. It is the file that *extendcp.bat* looks for when prompting for CD2.

The copy line in *extendcp.bat* will reflect the aggregate installation location defined by *agginst.bat*, for example:

```
xcopy D:\EXTEND\ANDC C:\AggDir /s /e /i /h /y /q
```

Install.bat This batch file may be started more than once if chained reboots are involved. The final invocation of *install.bat* will call the *Activate.bat* batch file to continue the IUI process.

If you do not require a chained installation, set ChainRBSRequired="NO"

```
:SETTYPEINSTALLATION
set ChainRBSRequired="NO"
```

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

As *install.bat* is started more than once, it needs a marker to count the invocations. Add the following lines to ensure *install.bat* creates a marker on the first invocation and detects its presence on the second run.

```
REM Install Software
if exist C:\INSTALL\ADDINST.CHK goto DECIDE
echo "AGGINST started" > C:\INSTALL\AGGINST.CHK
call C:\Install\IUIEcho Switch to INST1.bat...
```

APTRA Security Batch Files Refer to the APTRA Security release bulletin for the updates to the IUI batch files to enable auto-logon and start the applications.

The *s-bat.bat* batch file is used to start the application after reboot and automatically log on to SSTAuto1 user.

Add the following two lines for the Advance NDC application start procedure:

```
CD C:\ssds\dll
START STARTAPPS.VBS
```

This will enable the SST to boot up to SSTAuto1 and start the applications.

Update STARTAPPS.VBS The default *startapps.vbs* script provided with Advance NDC may need to be modified where super-aggregates are used to deliver APTRA components, as follows:

- If the APTRA XFS aggregate is not in your super-aggregate; for example, you have included only Self-Service Support
- If you include APTRA XFS but change its name
- If you change the default installation location for APTRA XFS

In these situations, *startapps.vbs* must be updated to reflect the changes, otherwise it will not be able to detect which version of APTRA XFS is installed.

The updated *startapps.vbs* can be delivered using the *custom.ini* method. For details of the *custom.ini* file, see “The Custom.ini File” on page 10-19.

Burn the CD. The CD needs to be created such that the Windows installer can read and copy the installed files to the hard drive.

The CD-ROM image must be of type ISO 1, but with the following options relaxed:

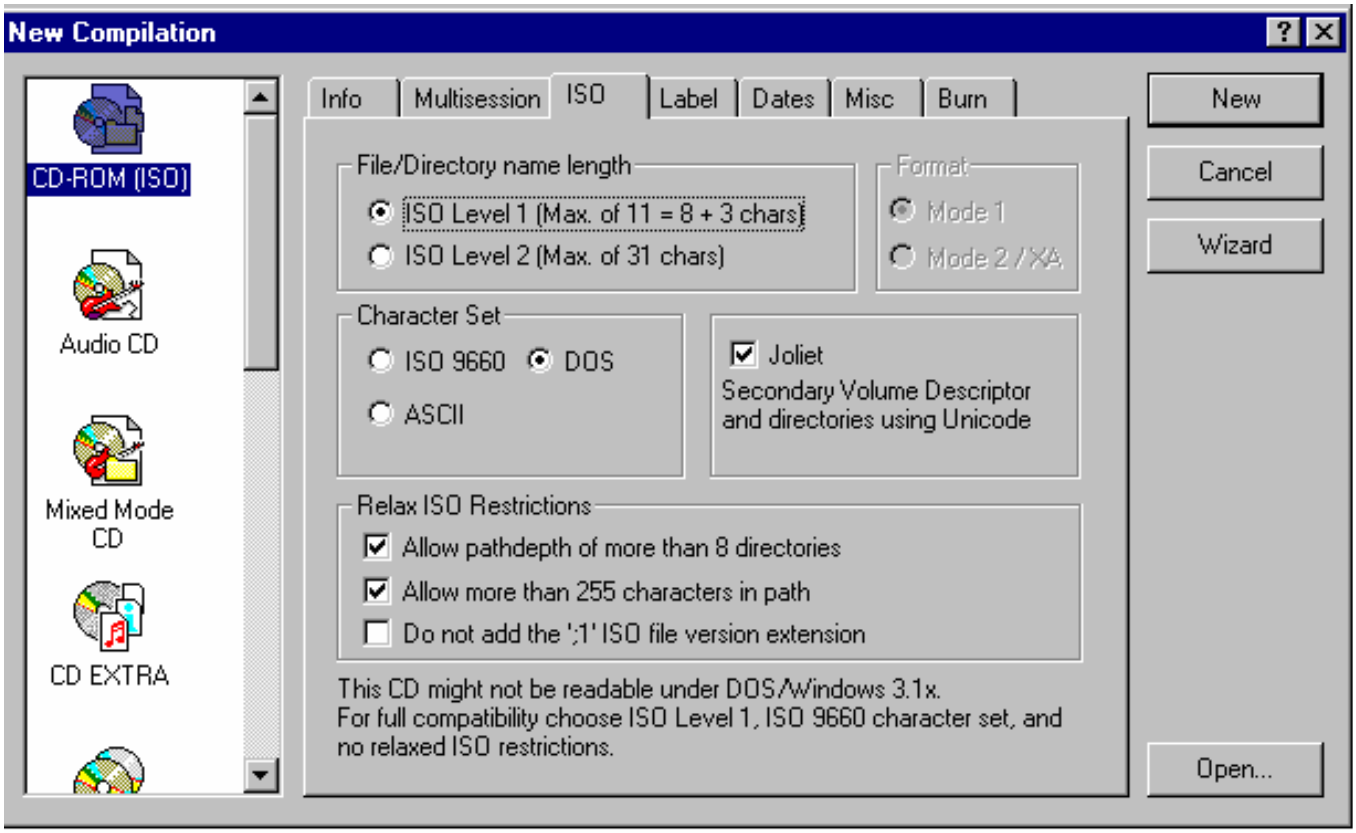
- Path depth must be allowed to be greater that 8 directories
- Path names must be allowed to be longer that 255 characters

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Ensure the Character Set is set to DOS and the Joliet option is checked.

Settings that work with NERO Burning ROM software are as follows:

Figure 10-2
NERO Burning CD ROM Settings



Steps in the IUI Process. For details of the IUI process, refer to the APTRA on-line documentation under **Initial Unattended Installation**.

Advance NDC Specific Steps. Complete the following steps for Advance NDC:

1 Create a super-aggregate for the installation of APTRA XFS and Advance NDC on an SST.
See “Prepare the Advance NDC Super-Aggregate.” on page 10-7 for details.

2 Include the aggregate in the IUI.
See “Prepare the CD Layout.” and “Update the IUI and Security Batch Files.” on page 10-8 for details.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

3 Create the CD-ROM.
See “Burn the CD.” on page 10-9 for details.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

NCR Encryptor Configuration

For the Advance NDC application, the following configuration is required for the NCR encryptor on NCR SSTs:

- Registry setting for the Encrypting PIN Pad (EPP) or Basic Alpha PIN pad and Encryptor (BAPE) device.

  The setting must be updated manually if the device is changed from EPP to BAPE or BAPE to EPP.

- Key names, which are automatically set up by the Key Manager utility.

Registry Setting for EPP Devices

If the NCR SST has an EPP device, the following registry key setting is 1 to detect the EPP device before starting the PIN SP.

```
HKLM\SOFTWARE\NCR\XFS PIN Service Provider\
XFS-DeviceControl\PIN\GENERAL_CONFIGS\EPPBTripleDESMode
```

When the PIN SP is started, it will change the other registry settings and load the appropriate libraries for the EPP.

If you subsequently change the device from EPP to BAPE, you must change this setting to 0.

Registry Setting for BAPE Devices

If the NCR SST has a BAPE device, the same registry key is used as for an EPP device, but the value is 0 to detect the BAPE device before starting the PIN SP.

If you subsequently change the device from BAPE to EPP, you must change this setting to 1.

Setting Basic or Enhanced Mode

The PIN capabilities are used to distinguish between basic and enhanced remote key loading. For further information, refer to the *APTRA Advance NDC, Reference Manual.*

The following registry key is used to select the remote key loading type:

```
HKLM\SOFTWARE\Classes\WOSA/XFS_ROOT\SERVICE_PROVIDERS\PIN\
GENERAL_CONFIGS\UseEnhRemoteKeyLoad
```

Valid values are as follows:

- 0 to select basic remote key loading. This is the default
- 1 to select enhanced remote key loading

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Key Names

As XFS requires keys to be created before loading them into the device, the NCR PIN SP provides registry settings for this purpose on NCR SSTs. The registry setting to run the Key Manager utility is automatically set when the application is first run, but it must be updated manually thereafter if the device is changed.

Key Manager Functionality

When Advance NDC is first run after installation on an NCR SST, the Key Manager utility detects the registry settings and initialises the keys of the physical encryptor with the default registry values.

When the Key Entry Mode is changed, the multi-vendor application automatically updates the registry settings and runs the Key Manager utility.

For other situations such as upgrading the software or changing the encrypting device from BAPE to EPP, the registry settings have to be checked and updated manually if required. See “Retaining Encryption Keys” on page 10-15 for details.

Note: If the registry setting is not correct for the device in use, unpredictable behaviour will result. If this occurs, update the registry key with the correct setting and run the application again.

Registry Settings for Key Manager

The following registry key is used:

```
HKLM\SOFTWARE\NCR\Advance NDC\ENCRYPTORMODE
```

The Encryptor Mode key has two settings as follows:

- EncMode
- KeyMan.

EncMode EncMode can have the following values:

Table 10-3
EncMode Values

Value	Meaning
1	Run Key Manager for BAPE or EPP in single-length mode
2	Run Key Manager for EPP in single-length XOR mode
3	Run Key Manager for EPP in double-length mode
4	Run Key Manager for EPP in restricted mode

The default value is 1.

All key modes (1 - 4) are supported for non-secure EPP on NCR and other vendors’ machines. With secure EPP and International

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Security (DAPI7) only key mode 4 is supported, if a key mode other than 4 is entered, the key load will fail.

KeyMan KeyMan can have the following values:

Table 10-4
KeyMan Values

Value	Meaning
0	Don't create the keys
1	Create the keys based on the value of EncMode. *See* Note 1.
2	Migrate keys from the previous versions

The default value is 2.

KeyMan is set to 0 after the Key Manager utility has successfully run. This prevents the Key Manager utility from running each time you start the application.

Note 1: Setting KeyMan to 1 deletes the keys from the *krep.dat* file and reinitialises the encryptor based on the EncMode registry setting.

Changing the Encryptor—Scenarios

This section summarises the procedure for changing the encryptor type.

Changing from EPP to BAPE

1 Check that the XFS Registry key is set for the BAPE encryptor, as follows:

```
EPPBTripleDESMode = 0
```

2 Set the Key Manager registry settings as follows:

```
EncMode = 1
KeyMan = 2
```

3 Key manager will start automatically while running the applications

Changing from BAPE to EPP

1 Check the XFS registry key is set for the EPP, as follows:

```
EPPBTripleDESMode = 1
```

2 Set the Key Manager registry settings as follows:

```
EncMode = 1
KeyMan = 2
```

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Note: With International Security installed, Advance NDC automatically sets the EncMode to 4 regardless of the setting entered here.

3 Key Manager will start automatically while running the applications.

Retaining Encryption Keys

Wherever possible, Advance NDC will retain the encryption keys used previously and use them in the new installation, or upgrade, of Advance NDC.

Note 1: If the hard disk already has a *krep.dat* file that does not match the Mode set in the registry, then the application will generate an exception. In this case, delete *krep.dat* and *krep.tmp* as described in "Troubleshooting Key Manager" on page 10-16, and run the applications again. This situation happens when the hard disk has been used on another encryptor with a mode different from the one selected.

Note 2: If the mode selected does not match the mode available in the encryptor, atypical errors occur; for example, when performing a key exchange operation, incorrect key data will be seen.

The following information applies to APTRA XFS 04.05.xx.

BAPE

To retain keys for migrating a BAPE, or EPP in BAPE emulation mode, do the following:

1 Set the XFS registry key, as described in "Registry Setting for BAPE Devices" on page 10-12.

2 In the registry, set the following values:
- EncMode = Previous mode (1 or 2), or leave unspecified. If specified, the previous mode is retained. If the value is unspecified, it is automatically set to 1 as default.
- KeyMan = Leave unspecified. If the value is unspecified, it is automatically set to 2 as default. Setting KeyMan to 1 deletes the keys from the *krep.dat* file and reinitialises the encryptor based on the EncMode registry setting.

3 Either retain the *krep.dat* file, or install an empty *krep.dat* file.

Basic Security (DAPI1)

To retain keys for migrating an EPP with key modes 1, 2, or 3, do the following:

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

1 Set the XFS registry key, as described in "Registry Setting for EPP Devices" on page 10-12.

2 In the registry, set the following values:
   — EncMode = Previous mode (1, 2, or 3), or leave unspecified. If the value is unspecified, it is automatically set to 1 as default.
   — KeyMan = Leave unspecified. If the value is unspecified, it is automatically set to 2 as default. Setting KeyMan to 1 deletes the keys from the *krep.dat* file and reinitialises the encryptor based on the EncMode registry setting.

3 Either retain the *krep.dat* file, or install an empty *krep.dat* file.

International Security (DAPI7)

Key mode 4 is for use with International Security, which has been supported from Advance NDC 3.0. Therefore, this section does not apply if you are migrating from an earlier release of Advance NDC.

To retain keys for migrating an EPP in key mode 4, do the following:

1 Set the XFS registry key, as described in "Changing from BAPE to EPP" on page 10-14.

2 In the registry, set the following values:
   — EncMode = 4. The mode cannot be retrieved from the encryptor in secure mode and, therefore, must be specified here.
   — KeyMan = Leave unspecified. If the value is unspecified, it is automatically set to 2 as default. Setting the KeyMan value to 1 deletes the keys from the *krep.dat* file and reinitialisaes the encryptor.

3 Retain the *krep.dat* file.

Troubleshooting Key Manager

If a power failure occurs while running Key Manager, or you encounter other problems with it, complete the following:

1 Close the Advance NDC application.

2 Delete the following files:
- *krep.dat*
- *krep.bak*

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

The default location for the *krep.** files is *C:\Program Files\NCR APTRA\XFS PIN Service Provider*. If the location is different, the directory path is available in the `Repository` registry setting under the following key:

```
HKLM\SOFTWARE\NCR\XFS PIN Service
Provider\XFS-DeviceControl\PIN\KEYLIB
```

3 Ensure the registry settings for the Key Manager are correct for the attached BAPE or EPP device (see "Registry Settings for Key Manager" on page 10-13).

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Preparing a Modified Advance NDC Aggregate for Installation

From Release 3.01, Advance NDC is supplied as a single aggregate containing all Advance NDC components and the Component Definition Tool (CDT) is not included.

Before modifying the installation in any way, ensure that you have performed all the necessary changes, as discussed either in "Changes Required" in Chapter 3, "Migrating Existing NDC+ Applications to Advance NDC" or in Chapter 4, "Upgrading from Earlier Releases of Advance NDC".

To re-use the product components to create a subset or a superset (including, for example, the SNMP Agent) of the supplied aggregate. See "Creating Aggregate Subsets and Supersets".

You use the *custom.ini* file to include customised file, registry settings and DLLs you have developed, such as worker class DLLs or external C function DLLs, see "The Custom.ini File" on page 10-19.

To extend the product components with additional file and registry settings, see "Extending the Supplied Application" on page 10-19.

Creating Aggregate Subsets and Supersets

If you modify the Advance NDC application using the Author, you use the Author to build the component, and the ABT to create and export an installable aggregate.

The provided aggregate can be modified using the Aggregate Builder Tool (ABT) on the development PC as follows:

1. On the development PC, create a new empty archive using **File | Select Archive**.

2. To populate this archive, import the aggregate from the APTRA Advance NDC Package CD using **File | Import**. You have only to select the path where an aggregate or component is located.

3. Add, edit and remove components as required. For example, you can add the 'SNMP Agent for APTRA' component to the aggregate.

   Note: It is not possible to update the APTRA Security component on an SST on which it is already installed. To install a newer version of the APTRA Security component the options are disk-replacement, disk-imaging or Initial Unattended Installation.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

4 Save the archive and then perform an export to produce an executable installable aggregate containing your changes, ready for the SST.

Tools Component

The Tools component will not be installed in a Runtime SST installation.

You can create a separate aggregate for the Tools component if you want to install it on an SST.

This aggregate would then need to be installed using the Development and Simulation option.

Extending the Supplied Application

If you have new files or registry settings to update the supplied application, use the custom.ini file to include them. All registry entries and source and target file names are held in the *custom.ini* file. Refer to "The Custom.ini File" on page 10-19 for details.

Complete the following steps:

1 Create a custom directory and copy the component files to it.

This folder can be created anywhere on your system, however, it must contain the *ANDC.msi* and *_comp.ini* files. The ABT recognises any folder containing *_comp.ini* as an APTRA component folder.

2 Create the *custom.ini* file, specifying the required registry and file entries as shown in Appendix F, "Example Custom.ini Files", and save it to the custom directory.

3 Copy all the files referenced in the *custom.ini* file to the custom directory.

4 Use the ABT to produce an installable aggregate. Refer to the on-line help provided with the tool.

The *Custom.ini* File

You must create a *custom.ini* file to specify the registry and file entries for a modified component.

The registry and file entries in this file can be associated with a particular configuration set, from which profiles are created in the ABT. You must edit the *custom.ini* file directly to add a new configuration set. This is because configuration sets cannot be duplicated using the ABT as no configuration DLL is defined for Advance NDC.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

The files specified in the *custom.ini* file must be in the components folder at installation. The default target path for a file, if it is not specified, is [INSTALLDIR]. [INSTALLDIR] is the directory chosen during installation as the target installation folder. For SST installations, this is *C:\ssds*. For development installations, the default is *<drive>:\ntglobal*, but this can be changed at installation.

To use another target path, include the full path, for example, *C:\MyFolder\File1.dll*. To place a file in a subdirectory, specify the full target path, for example, *[INSTALLDIR]\MySubFolder\File2.dll*, or *C:\MyFolder\MySubfolder\File3.dll*.

A prefix must be used to specify the data type when defining a registry value in the *custom.ini* file. String values are the only exception. The data types and associated prefixes are shown in Table 10-5.

Table 10-5
Registry Data Type Prefixes

Data Type	Prefix	Example
REG_SZ		HKEY_LOCAL_MACHINE\Software\Subkey\StringValue1=MyString
REG_DWORD	#	HKEY_LOCAL_MACHINE\Software\Subkey\NumberValue1=#1
REG_BINARY	#x	HKEY_LOCAL_MACHINE\Software\Subkey\BinaryValue1=#x010203
REG_MULTI_SZ	[~]	HKEY_LOCAL_MACHINE\Software\Subkey\MultiStringValue1=[~]str1[~]str2

Installing the Aggregate

The aggregate can now be used in an installation. The following list provides examples of installation methods:

- Interactively by running *setup.exe*, located in the exported aggregates folder, and selecting the target type (Development and Simulation or SST) and profile required
- Silently for the development target type and no profile. To do this, run the following command line from the export aggregate's root directory:

  ```
  setup.exe /u:u /t:devsim
  ```
- Silently for the SST target type and a named profile. To do this, run the following command line from the export aggregate's root directory, in this example we are using a profile named ProfileForSet1:

  ```
  setup.exe /u:u /t:sst /f:"ProfileForSet1"
  ```

Deinstalling the Aggregate

To deinstall the aggregate silently, run the following command line from the aggregate installer's installation folder:

```
setup.exe /d:"application name" "version number" /u:u
```

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

To deinstall the aggregate interactively, use the Add/Remove Programs option in the Windows Control Panel.

Post-Installation Activities

To set up the SST, complete the following steps:

1 If you modify the registry settings for customisations, they will need to be reset. Installing Advance NDC sets all registry keys to the default.

2 Start or activate the Advance NDC application as follows:

   a Ensure the Supervisor Mode switch is set to Supervisor, to enable all communications, keys, message mode and MAC to be correctly set up before a connection to the host is opened.

   b Select **Start | Programs | NCR APTRA | Advance NDC | Start Advance NDC**.

3 Complete the following in Supervisor before performing any transactions:

   a Enter the encryption keys.

   b Clear and set the note counts.

   c Configure communications.

   For details of setting up an SST using Supervisor, refer to the *APTRA Advance NDC, Supervisor's Guide*, and if you are setting up SSTs on other vendors' SSTs, the *APTRA Advance NDC, Multi-Vendor Support Reference Manual*.

Using TCP/IP to communicate with Central

You must specify the IP addresses and port numbers using Supervisor.

For details of the TCP/IP Configuration menu, refer to the *APTRA Advance NDC, Supervisor's Guide*.

For additional communications configuration, see "Configuring Communications" on page 5-11.

For details of the Communications Event Log, refer to the *System Application User Guide*.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and/or distribution is strictly prohibited.

De-installing Advance NDC

If an SST has a version of Advance NDC prior to version 3.01 or you want to re-install Advance NDC, you must remove the existing installation and any associated software that has been installed separately before installing Advance NDC.

If you re-install Advance NDC to the same location, it overwrites most of the contents of the Advance NDC directories, so you need to take copies of any files you wish to keep. Complete the following steps:

1. On the PC, if you have carried out any development under the root directory of Advance NDC (default name is *<drive>:\ntglobal*), or have changed any files under it, back up the development or changed files.

2. On the PC or SST, back up any electronic journal or hard copy backup logs in the *C:\Program Files\NCR APTRA\Advance NDC* directory.

3. In the Windows Control Panel, use **Add or Remove Programs** to first remove any Advance NDC service pack, then the Advance NDC components. Files created after the installation of Advance NDC, such as *custom.dat*, user ID files and CDI data files, will not be deleted. It is up to you to keep or delete them.

Note: If you are deinstalling Advance NDC because the EJ Privacy password needs to be reset, you must also delete the UCDI persistent file. For further information, see “EJ Privacy” on page 5-40.

4. Install Advance NDC.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and/or distribution is strictly prohibited.

Chapter 11

Troubleshooting

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Overview

This chapter provides the following information on troubleshooting with Advance NDC:

- An overview of the available troubleshooting tools and their application
- An introduction to Problem Determination event and error logs
- Trace information and methods of introducing trace points in Advance NDC
- Using Silent Debug
- Using DebugLog.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Advance NDC Troubleshooting Utilities

In Advance NDC the following troubleshooting utilities are available:

- Problem Determination (PD)
- Silent Debug
- DebugLog.

Selecting a Debugging Utility

Choosing the most suitable utility for your troubleshooting investigation depends on the level and type of problem that requires investigation.

Problem Determination

Use PD if you need to analyse data specified by a template. Templates are supplied with PD and Advance NDC, or you can create your own.

Details of templates are provided in the APTRA on-line help under **Problem Determination**.

Problem Determination collects a large amount of data in a short period of time. It is not suitable for collecting data over extended periods because of the amount of data involved.

For more information about using PD with Advance NDC see “Troubleshooting with Problem Determination” on page 11-9.

Silent Debug and Debug Log

The same information is logged when using the Silent Debug or DebugLog tool, but they are designed for use in different environments as follows:

- Use Silent Debug in a live environment.
- Use DebugLog in a test environment.

Note: If you have both Silent Debug and DebugLog installed, they cannot be run at the same time. If an attempt is made to run one after the other has been started, a message will be logged in the event log stating which one could not start.

Silent Debug saves the data to log files. DebugLog displays the information on screen. DebugLog menu options can be used to save the logs manually or it can be configured to save to log files

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

automatically. For information on configuring automatic saves see “Configuring Auto Save” on page 5-69.

The log files are significantly smaller than those collected by PD allowing you can collect data over a much longer period without file storage problems.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Event and Error Logs

For troubleshooting with PD the following Windows event logs are available:

- Application event log
- System event log
- Security event log

Almost all internal NCR APTRA XFS platform events are written to the application event log.

Windows XP events are written to the system event log.

Saving Windows event logs is described in the *Self-Service Support, System Application User Guide*.

The Unhandled Exception Handler (UEH) provided with APTRA XFS records the same exception information in both the Windows application event log and the UEH log, but the entries in the UEH log are easier to read.

The UEH log file (*ueh.log*) is located in <*pathname*> *system32*> where <*pathname*> is the installation directory for Exception Handling. The last exception is always at the end of the file. The way to check the UEH log file is to go to the end of the file and search backwards for the word 'exception'.

The APTRA XFS platform also provides a software log, communications log and device log. The information in these logs is also contained in the Windows application event log.

For details of the UEH refer to the APTRA on-line documentation under **Exception Handling | About Exception Handling | Exception Handling Functions | Unhandled Exception Handler**.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Advance NDC Trace Information

Advance NDC has tracing functionality built into various points in the application (known as trace points). You can add more trace points if necessary. When Advance NDC reaches a trace point, it will send a trace output to the active debug tool (Silent Debug or DebugLog).

The Advance NDC application is the sender (client) of the trace information and Silent Debug or DebugLog is the receiver (server). A timestamp is added to the logged trace information to help in identifying what was happening when a problem occurred.

The trace information is categorised into named trace streams. The default trace streams used by Advance NDC are described in Table 11-1.

You can add new trace streams if required, as described in the following sections:

- "Tracing from C++ Code".
- "Tracing in the Author"

Table 11-1
Default Trace Stream Types

Trace Stream Name	Description
MESSAGEIN	Logs all messages received by the SST
MESSAGEOUT	Logs all messages sent from the SST
OOXFS	Logs interaction between Advance NDC and the CEN-XFS sub-system
DEBUG	Logs the general trace stream from the trace points in Advance NDC. This is the trace stream that is usually investigated first
STATEDATA	Logs the NDC state flow
SECURITY	Logs security-related information

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Tracing from C++ Code

To trace information from C++ code, create an instance of the `DebugOut` class and use the `debug()` method.

The debug method accepts a variable number of parameters organised in the `printf()` style. The code needs to include `debugout.h` and link with *DebugOut.lib*. The pre-processor symbol DBON must be defined in the Visual C++ (IDE) project settings. This dynamically loads *ssdsDebugOut.dll* at runtime.

To suppress debug messages, you can use the following macro:

```
DBG(dbg.debug("Won't Print Out Unless DBON is defined");)
```

In Figure 11-1, the "Hello World from NCR" message is traced. This time, the company name is dynamically supplied as a parameter in the `debug()` method using the `printf()` output formatting style.

Figure 11-1
Example Tracing Script

```
#define DBON
#include "debugout.h"
void HelloWorld()
{
     char * CompanyName = "NCR";
     DebugOut dgb("DEBUG");
     dgb.debug("Hello World from %s", CompanyName);


}
```

Tracing in the Author

To implement tracing in the Author, edit the Author flow as described in the following sections:

- "Using the Debug Message Sender"
- "Using Active Script Hosts".

Using the Debug Message Sender

To implement tracing in an authored application, the Debug Message Sender worker is added to the Author flow as illustrated in Figure 11-2.

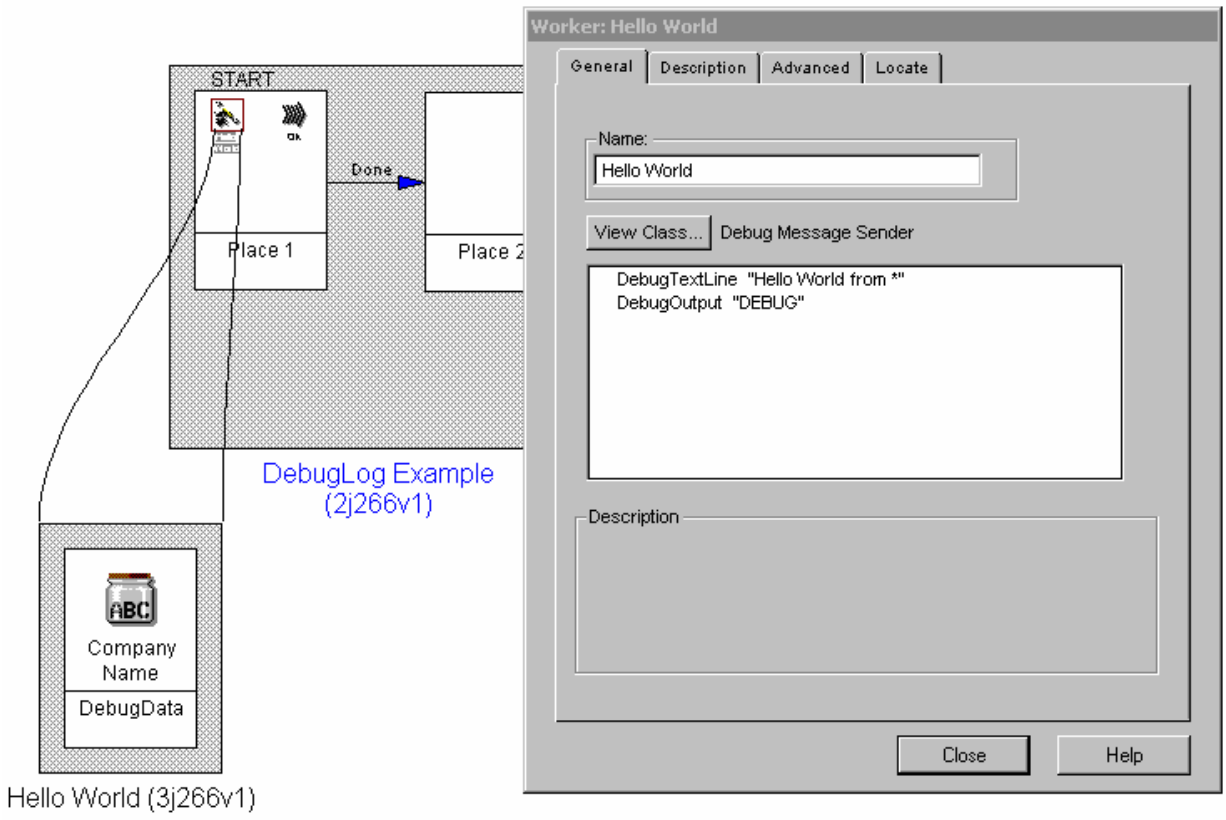
Trace information is specified in the `DebugTextLine` attribute. If the text line contains an asterisk ('*'), dynamic contents retrieved from the worker's workgroup are inserted in the text in that position. For details of Message Sender worker refer to the worker class on-line help provided with the Author.

The name of the trace stream is specified in the worker settings using the `DebugOutput` attribute.

In Figure 11-2, the "Hello World from NCR" message is traced in the `DEBUG` trace stream.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Figure 11-2
Tracing in the Author Flow



Using Active Script Hosts

To add debugging to a new or existing Active Script Host worker, create an instance of the `DebugMessage.DebugMessage` object and use the `debug(String text)` method.

In Figure 11-4, the “Hello World from NCR” message is traced using an Active Script Host worker.

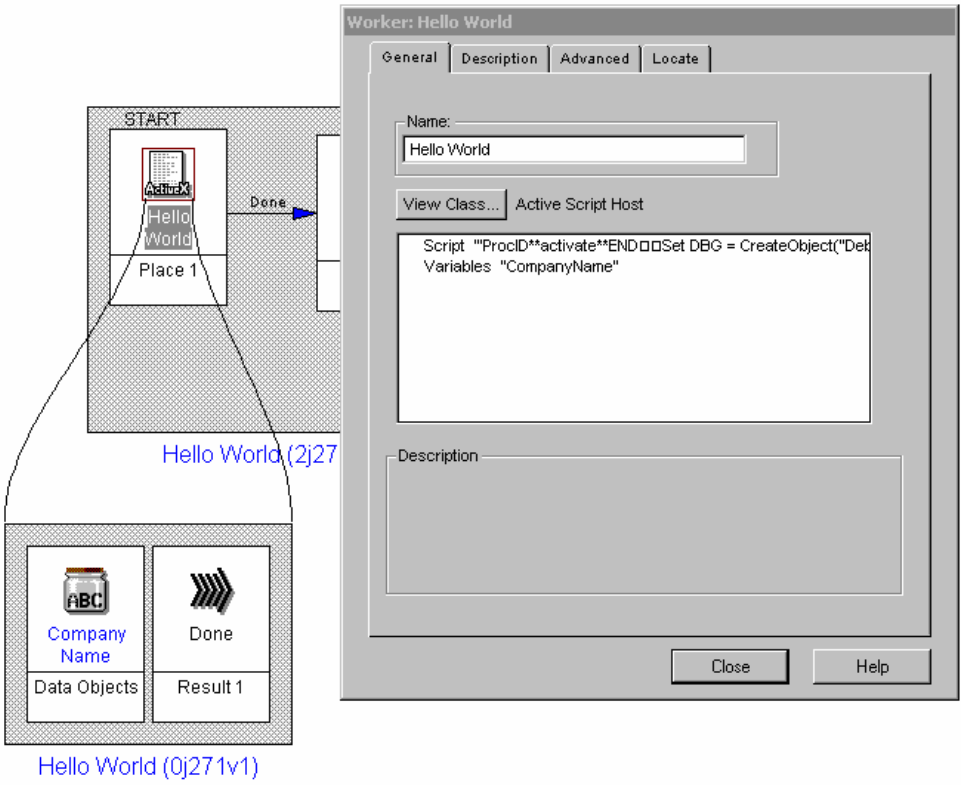
The script illustrated in Figure 11-3 is added to the `Script` attribute.

Figure 11-3
Example Tracing Script for the Active Script Host Worker

```
Set DGB = CreateObject(“DebugMessage.DebugMessage”)
Sub activate()
  On Error Resume Next
  Dim text
  text = “Hello from “ + CompanyName.Data
  DBG.DebugOutput = “DEBUG”
  DBG.debug(text)
  ActiveWorkGroup.Index = 1
End Sub
```

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Figure 11-4
Tracing with an Active Script Host Worker



Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Troubleshooting Tools

This section provides an overview of Problem Determination, information on using the Silent Debug and DebugLog tools and a description of their log files.

Troubleshooting with Problem Determination

The Problem Determination (PD) components, which are provided as part of the Advance NDC aggregate, are as follows:

- Problem Determination Analysis (PDA) must be installed on the development PC.
- Problem Determination Collection (PDC) must be installed on the SST.

  Note: If an investigation has to be viewed on the SST, Problem Determination Analysis, which does not require the use of keyboard, must also be installed.

Advance NDC provides a default template file (*filterset.txt*) to help with PD investigations. This template enables the tracing of the information captured in the MessageIn, MessageOut, and OOXFS windows of DebugLog. The default template and the executable *makecab.exe*, which compresses the investigation files, are automatically delivered when Advance NDC is installed on an SST. If you modify or create a template, it must be configured as described in the APTRA online documentation under **Problem Determination**.

Note: The default location for the templates and the log file created by an investigation is in the PD installation folder; for example, if PD is installed on the C drive, the following are the default locations:

- *c:\program files\ncr aptra\problem determination collection\templates*
- *c:\program files\ncr aptra\problem determination collection\investigations*

Makecab.exe is located in *c:\program files\ncr aptra\pd* on the SST.

For information on using Problem Determination Collection and Analysis for troubleshooting refer to the APTRA on-line documentation under **Problem Determination.**

Troubleshooting with Silent Debug

Silent Debug provides trace logging in a live environment. When installed and started it creates trace log files, depending on the configuration settings in the *SilentDebug.ini* file. Installating and

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

configuring Silent Debug is described in “Silent Debug” on page 5-66.

Silent Debug can be controlled in the following ways:

- Using the Configure menu in Supervisor as described in the *APTRA Advance NDC, Supervisor’s Guide*.
- Using the command line
- Using a Windows script to start logging remotely.

Note: Windows Net commands are not supported.

Using the Command Line

Silent Debug is run using the following command:

```
NCRSilentDebug [option]
```

The options that can be used with Silent Debug are described in Table 11-2.

Table 11-2
Silent Debug Commands

Option	Description
-i	Installs Silent Debug as an operating system service and registers it with the Service Control Manager. This command installs the service in Automatic Restart Mode and in a stopped state.
-u	Uninstalls Silent Debug as an operating system service. This command un-registers the service with the Service Control Manager, but does not delete the *SilentDebug.ini* or *NCRSilentDebug.exe* files. Silent Debug must be stopped before running this command.
-v	Displays the version of Silent Debug.
-r	Starts Silent Debug
-s	Stops Silent Debug
-vl	Validates the *SilentDebug.ini* file
-st	Gets the current status of SilentDebug tracing

Starting the Silent Debug Service Remotely

Silent Debug can be started remotely using a Windows script that changes the value of a CDI store (Store ID: 3348).

Figure 11-5 is an example of a script used to start Silent Debug remotely.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Figure 11-5
Example Script to Start Silent Debug

```
Dim objSilentDebugStore
Set objSilentDebugStore = CreateObject("COMCDISTORE.COMCDIimpl")
On Error Resume Next
'1 - Integer Type
If objSilentDebugStore.setStoreId (3348,1) = 0 then
objSilentDebug.Data = 1

End if
```

Silent Debug Log Files

Silent Debug log files are created in the following location:

```
C:\Program Files\NCR APTRA\Advance NDC\Debug
```

The first line of the log file records the date in `DD/MM/YYYY` format, the time in `HH:MM:SS` format, and how many times the file has been overwritten as follows:

```
Trace Created 02/01/2006 15:06:44 Counter = 0
```

The counter indicates the number of times the file has been overwritten from the beginning. The count starts again at 1 after the counter reaches 99,999. If the file is being overwritten too frequently, increase the configured file size.

A marker, `<END>`, is written to the file at the end of the trace stream information. Any content appearing after the `<END>` marker is trace stream information that exists from the last time the file was written to.

See "Configuring Silent Debug" on page 5-66 for details of log file configuration.

Single Trace Stream Log Files Single trace stream log files record information from one type of trace stream.

Figure 11-6 is an example of a `MESSAGEIN` trace stream log file. Each line uses the following format, as described in Table 11-3:

```
HH:MM:SS.sss - DD/MM/YYYY #xxxx [xxxxxxxx]
```

Table 11-3
MESSAGEIN Line Format

Format	Description
`HH:MM:SS.sss`	Time of log entry in hours, minutes, seconds, and microseconds
`DD/MM/YYYY`	Date of log entry
`#xxxx`	Message sequence number
`[xxxxxxxx]`	The message that has been received For information on the message format, refer to the *APTRA Advance NDC, Reference Manual*

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Figure 11-6
Example MESSAGEIN Log File

```
        Trace Created 06/03/2007 15:57:11        Counter = 0
 15:57:11.821 - 06/03/2007   #0001   [12000P21]
15:57:14.836 - 06/03/2007   #0002   [12000P21]
15:57:17.852 - 06/03/2007   #0003   [12000P21]
15:57:20.883 - 06/03/2007   #0004   [12000P21]
15:57:23.899 - 06/03/2007   #0005   [12000P21]
15:57:26.914 - 06/03/2007   #0006   [12000P21]
15:57:29.930 - 06/03/2007   #0007   [12000P21]
15:57:32.961 - 06/03/2007   #0008   [12000P21]
15:57:35.977 - 06/03/2007   #0009    [12000P21]
15:57:38.993 - 06/03/2007   #0010   [12000P21]
15:57:42.008 - 06/03/2007   #0011   [12000P21]
16:02:29.901 - 06/03/2007   #0012   [12000B0000]<END>
```

Merged Trace Stream Log Files Merged trace stream log files record information from all of trace streams.

Each trace stream entry is identified by opening and closing tags, indicating the type of trace stream content. For example, any information from the MESSAGEOUT trace stream is enclosed in <MESSAGEOUT> and </MESSAGEOUT> tags.

Figure 11-7 is an example of a merged trace file which contains trace stream information for `DEBUG` and `MESSAGEOUT`.

The `DEBUG` lines contain the following:

- Opening tag to identify trace stream `<DEBUG>`
- Current mode
- Time of log entry
- Description of activity
- The worker identifier contained in `<wid>` and `</wid>` tags. For example, in Figure 11-7 `<wid>22v1823m34</wid>` identifies the Debug Message Sender worker.
- Closing tag to identify trace stream `</DEBUG>`.

The `MESSAGEOUT` lines contain the following:

- Opening tag to identify trace stream `<MESSAGEOUT>`
- Time of log entry
- Date of log entry
- Consecutive order of sent message
- The message details
- Closing tag to identify trace stream `</MESSAGEOUT>`.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Figure 11-7
Example Merged Trace File

```
        Trace Created 06/03/2007 15:53:32        Counter = 0
<DEBUG>Supervi.. 15:57:11.774 Activating Pending Supervisor
Entry Msg = 12000P21
Exit Msg =   <wid>6r641ichl21</wid>
</DEBUG>
<DEBUG>Supervi.. 15:57:11.774 Attempt to send Pending Supervisor Entry Message  <wid>22v1823m34</wid>
</DEBUG>
<MESSAGEOUT>15:57:11.821 - 06/03/2007   #0001   [12000P21]</MESSAGEOUT>
<DEBUG>Supervi.. 15:57:11.821 Supervisor Pending Complete  <wid>25v1823m34</wid>
</DEBUG>
<DEBUG>Supervi.. 15:57:14.821 Activating Pending Supervisor
Entry Msg = 12000P21
Exit Msg =   <wid>6r641ichl21</wid>
</DEBUG>
<DEBUG>Supervi.. 15:57:14.836 Attempt to send Pending Supervisor Entry Message  <wid>22v1823m34</wid>
</DEBUG>
<MESSAGEOUT>15:57:14.836 - 06/03/2007   #0002   [12000P21]</MESSAGEOUT><END>
```

Troubleshooting with DebugLog

DebugLog provides trace stream logging in a test environment. The trace information is displayed in windows and can be saved to file automatically or manually. For information on installing and configuring DebugLog, see “DebugLog” on page 5-69.

To start DebugLog select **Start | All Programs | NCR APTRA | Advance NDC | Tools | DebugLog.**

The trace stream windows can be scrolled through, or saved. For information on configuring save options, see “Configuring Auto Save” on page 5-69 and “Changing the Save All Destination” on page 5-70.

Trace Stream Information

A window containing the trace stream messages for each trace stream type is displayed in the DebugLog main window. To view all trace stream windows simultaneously, select **Tile** from the **Window** menu.

Note: Field separators can be displayed in various ways as they are unprintable characters. For example, square boxes or ‘L’ shapes can be used on-screen and in files. In the *APTRA Advance NDC, Reference Manual*, FS is used to represent field separators in Chapters 9 and 10, and a tilde (~) is used in Appendix D.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Chapter 12

Modifying the Advance NDC Applications

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited

Overview

This chapter describes the various methods of modifying Advance NDC and discusses what is best suited for different scenarios. Which method you choose depends on the nature of the change you plan to make, your experience and how much re-implementation will be required for future releases of Advance NDC.

This chapter also considers the following:

- Whether development is necessary
- The different implementation techniques available

For more information about using the different kinds of Exits to enhance or extend Advance NDC, see Appendix A, “Authored Exits”.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Ways of Modifying Advance NDC

Before starting to implement your particular modification to Advance NDC, establish which of the following possible approaches applies to you:

- Using an existing implementation
- Modifying an existing implementation
- Creating a new implementation.

Use an Existing Implementation

If Advance NDC contains all the functionality required for a particular customer environment, you can use Advance NDC as supplied by NCR. You will, however, still have to perform the usual NDC+ modifications as follows:

- Define consumer transactions (create the appropriate states to define the consumer flow, define consumer screens, FITs to define card handling rules and so on)
- Define configuration parameters to control system behaviour
- Decide the configuration of the Supervisor when the application is deployed and running.

These changes are regarded as customisations of Advance NDC and are described in the *APTRA Advance NDC Reference Manual* and the *APTRA Advance NDC Supervisor's Guide*. This type of modification is not discussed in this chapter.

Modify or Create an Implementation

Modifying an existing implementation or creating a new implementation requires programming skills to modify an existing routine or write your own routine. These changes are regarded as enhancing and extending Advance NDC.

For various enhancements or extensions to Advance NDC, different programming techniques are available. They each require different programming skills. Writing your own routine, called a C Exit, requires knowledge of C or C++, but it can be used in future releases of Advance NDC without any or very little re-implementation. Modifying the Advance NDC authored applications is easier but the work will have to be re-implemented when you move to a future release of Advance NDC.

Before making programmatic changes consider the following:

- Will the functionality you want to add be available in a future release of the product and in time for your requirements?
- Professional Services may have an implementation that could meet your requirements.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

- If you are a subscriber to the Advance NDC news group, you can post a request for anyone who has implemented a similar, if not identical, enhancement to send it to you. Even if it does not provide exactly what you are looking for, it may be an excellent starting point.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Programming Techniques

The two main techniques for enhancing or extending Advance NDC are as follows:

- Writing C Exits (DLLs accessed through a specific exit interface)
- Using the APTRA Author to modify the authored flow using one of various techniques, some of which will require less re-implementation in future releases of Advance NDC.

C Exits

C Exits are called at predetermined points in the application flow. If these predetermined points are sufficiently flexible for your modification, then a C Exit is likely to be the most appropriate option.

Configuration files allow you to link up the C Exits you have developed with Advance NDC. There are a number of APIs available for access to Advance NDC functionality.

If keyboard and screen handling are required, a C Exit can be developed in the Author to take advantage of the support for keyboards and screens and the predefined infrastructure.

For more information about using C Exits in Advance NDC, see "C Exits" in Appendix A.

Using the Author

The Author provides a graphical approach to software development, with a diagrammatic representation of the application flow.

Although using the Author can save time during the initial implementation, a major disadvantage is the re-implementation effort required whenever you install service packs or move to a later release of Advance NDC.

Various worker classes are provided to offer predefined functionality and these can be used to create an application flow. A list of workers classes developed for Advance NDC is given in Appendix B, "Advance NDC Workers Supplied". The worker class help in the Author describes all workers. The *APTRA Author User's Guide* lists the workers supported in Advance NDC.

There are different methods available to add missing functionality. These methods can be considered implementation techniques in their own right and are the following:

- Script host
- External C function

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

- New worker class
- Web exit

Script Host

You can use the Active Script Host worker class to make use of COM objects and write VBScript to define the behaviour of the Active Script Host worker.

As Active Script Host workers must be active to process events, care is required when authoring a flow to ensure events are always picked up and processed when they should be.

If your application flow will interface to a number of COM objects, such as APTRA ActiveXFS objects, consider another implementation technique rather than using Active Script Host workers to drive the flow. For example, you can implement the application flow, or a part of it, in Visual Basic (VB) and invoke the VB COM object from a single Active Script Host worker containing minimal VBScript.

Note: There is a limit on the amount of script data supported within the Active Script Host, which may make this implementation technique inappropriate for your development.

For more information about ActiveX exits, see "ActiveX Exits" in Appendix A. For information about ActiveX controls, refer to the *APTRA Author User's Guide.*

External C Function

An external C function is different from a C Exit in the way it is called. You use an Exit Condition Tester or an Assigner worker to call the C function. The advantage is that a C function can be called anywhere in the authored flow.

The C function can contain any code that can be written using the Visual Studio 2005 compiler, provided the function is defined within a DLL and invoked according to the C calling convention.

Note: The authored flow will need to be updated to call the new C function, and if there is a change in the compiler and APIs used, the C function will have to be re-implemented.

For more information, see Chapter 5, "Configuring Advance NDC and Support Applications" and also refer to the *APTRA ADE Programmer's Guide.*

Note: The *APTRA ADE Programmer's Guide* has not been updated for the CEN-XFS interface.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Web Exits

If you want to implement an HTML-based flow, for example, for transactions not available on the NDC host, use the Web Exit state type for the call rather than the Web Page Loader. This will make for easier migration in the future.

For keyboard and display separation, use the Display and Keyboard hooks described in the white paper, *APTRA Advance NDC Display and Keyboard Hooks.*

New Worker Class

Developing a new worker class requires considerable effort and is not recommended if there is a suitable alternative technique that can be used instead. Consider whether you can combine existing workers within a director to perform the required task, or write a C Exit or external C function.

For more information, refer to the *APTRA Author User's Guide* and the *APTRA ADE Programmer's Guide.*

Further Reading

Chapter A, "Using Exits in Advance NDC" discusses the benefits of the different types of exit and when to use them.

The impact of multi-vendor functionality on C Exits is discussed in the white paper *NDC Exits and CEN-XFS in Advance NDC.*

The *APTRA Advance NDC Using NDC Exits* manual has a table in the preface which identifies the sections that are relevant for Advance NDC.

For general guidelines on developing Author flows, refer to the *APTRA Author User's Guide.*

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Implementation Options

This section identifies the best options for modifications to Advance NDC, from adding or modifying a device to selecting a code page.

You also need to consider what impact your modification will have on the message interface: device faults, configuration and supervisor functions; and how the impact can be avoided or minimised.

Adding or Modifying a Device

If you are considering adding a new device or modifying the functionality of an existing device, the first five implementation tasks in Table 12-1, "Common Scenarios for Enhancing or Extending Advance NDC" are typically required. You may have more tasks for your implementation but only the essential ones are described here: others are regarded as optional.

Scenarios for Enhancing or Extending Advance NDC

The first column in the table gives a typical modification, which may be part of a larger implementation such as adding a new device. The second and third columns give the recommended technique to use. References to further information, where appropriate, are given in round brackets. N/A means not applicable.

Table 12-1
Common Scenarios for Enhancing or Extending Advance NDC

Implementation Task	Create	Modify
State type	Depending on the functionality offered by the State Type: **C Exit** (*Using NDC Exits* manual, and Appendix A, "Using Exits in Advance NDC") **Author**, including updating the Customisation Layer (*Using NDC Exits* manual, and Chapter 7, "Enhancing the Customisation Layer")	**C Exit state types**, modify and re-release (*Using NDC Exits* manual, and Appendix A, "Using Exits in Advance NDC") **State types** that cannot be modified, create a new state type to supplement the existing one. **Authored state types**, update the Customisation Layer

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and/or distribution is strictly prohibited.

Implementation Task	Create	Modify
Terminal command	**Virtual controller,** to add extra data to an outgoing message, for example, EMV Exits (Chapter 3, "Migrating Existing NDC+ Applications to Advance NDC" and *Using NDC Exits* manual) **Author** to add a new terminal command (Chapter 8, "Enhancing the Application Core or Supervisor")	**Virtual controller,** modify and re-release **Author**, to add additional data to an existing terminal command for a new device
Status message	**C Exit**, use the appropriate Advance NDC API to send a new status message. (*Using NDC Exits* manual and *CEN XFS and NDC Exits in Advance NDC* white paper) **Authored flow**, use the NDC Message Sender worker to send a new status message **Virtual controller**, to append to status messages for existing functionality, use a C Exit send status message	**C Exit**, modify the API parameters to send a revised message, and re-release the C Exit. (*Using NDC Exits* manual and *CEN XFS and NDC Exits in Advance NDC* white paper) **Authored flow**, modify the NDC Message Sender worker to change the data sent to the host. For the status mapping tables used to generate the status messages built into the product, refer to the *APTRA Advance NDC Multi-Vendor Support Reference* manual.
Supervisor function	**C Exit** or **Author** depending on the functionality to be offered.	**C Exit,** if you have created a state type using a C Exit, modify and re-release the C Exit **Author,** if the Supervisor function is authored, update the Supervisor application **Supervisor screen content** without modifying the supervisor functionality or application flow (such as translating the screen text), edit the *resrvd.def* file released with Advance NDC. The file contains all the static text displayed on the screen when in Supervisor mode. (*APTRA Advance NDC, Reference Manual*)
Transaction request / reply	Transaction Request / Reply functionality already exists in the Customisation Layer application flow so you do not need to create transaction request and reply unless you are making significant changes to the Customisation Layer. **Author**, to create a new transaction reply function (Chapter 7, "Enhancing the Customisation Layer")	**Author**, to modify a Transaction Request / Reply *function*, modify the Customisation Layer application (Chapter 7, "Enhancing the Customisation Layer") **Virtual controller /C Exit**, to modify a Transaction Request / Reply *message* (*Using NDC Exits* manual)
COM object access without display	**Author**, use an Active Script Host worker	**Author**, modify an existing Active Script Host worker

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Implementation Task	Create	Modify
COM object access with display (non-Advance NDC window)	**Author**, use a Web Page Loader worker	**Author**, modify an existing Web Page Loader worker
Data Manipulation	**External C function**, if the data manipulation is complex (Chapter 7, "Enhancing the Customisation Layer", and the *ADE Programmer's Reference*) **Author**: — if interfacing to a COM object and / or the data manipulation is easier to write in VBScript, use an Active Script Host worker — if data manipulation is relatively simple and must be closely integrated into an authored flow, use Data Manipulation workers	As in the Create column but modify an existing worker or external C function
Start of Day processing	**C Exit**, using the Start of Day hook **External C function**, called by an Assigner worker within an Authored flow in the Customisation Layer application. Use this if the Start of Day hook is not located where you require. (Appendix A, "Using Exits in Advance NDC" and the *Using NDC Exits* manual)	**C Exit**, modify and re-release **External C function,** modify and re-release (Appendix A, "Using Exits in Advance NDC" and the *Using NDC Exits* manual)
Close State processing	**C Exit**, create using the Close state hook (Appendix A, "Using Exits in Advance NDC"; the *Using NDC Exits* manual)	**C Exit,** modify the C Exit (Close state hook) and re-release (Appendix A, "Using Exits in Advance NDC" and the *Using NDC Exits* manual)
Message processing (modify incoming and outgoing message content)	**C Exit**, (virtual controller) to modify the content of incoming and / or outgoing messages. (Appendix A, "Using Exits in Advance NDC" and the *Using NDC Exits* manual)	**C Exit,** modify the C Exit (virtual controller) and re-release (Appendix A, "Using Exits in Advance NDC"; the *Using NDC Exits* manual)
Supervisor Entry processing	**C Exit**, create using the Supervisor Entry hook. (Appendix A, "Using Exits in Advance NDC" and the *Using NDC Exits* manual)	**C Exit,** modify the C Exit (Supervisor Entry hook) and re-release (Appendix A, "Using Exits in Advance NDC" and the *Using NDC Exits* manual)
Supervisor Exit—about to exit processing	**C Exit**, create a C Exit and use the About to Exit Supervisor hook. (Appendix A, "Using Exits in Advance NDC" and the *Using NDC Exits* manual)	**C Exit**, modify the C Exit (About to Exit Supervisor hook). (Appendix A, "Using Exits in Advance NDC" and the *Using NDC Exits* manual)

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Implementation Task	Create	Modify
Supervisor Exit—exited processing	Create a C Exit (Exit Supervisor hook). (Appendix A, "Using Exits in Advance NDC" and the *Using NDC Exits* manual)	Modify the C Exit (Exit Supervisor hook). (Appendix A, "Using Exits in Advance NDC" and the *Using NDC Exits* manual)
Suspend Handling	**C Exit**, if you require suspend handling for a new device as the Advance NDC Suspend mode will not handle a new device	**C Exit**, modify and re-release
Journalling	**C Exit**, use the Advance NDC API PrintToJournal( ) to send the appropriate text to the journal printer. (*Using NDC Exits* manual) **Author**, for journal points within authored flows, create a new journal point in the appropriate application, depending on where the new journal point is needed; use a Journal Page worker to send the appropriate text to the journal printer. (Chapter 7, "Enhancing the Customisation Layer" and Chapter 8, "Enhancing the Application Core or Supervisor")	**Existing transaction journalling** can be configured through the Supervisor menus or the registry **C Exit**, modify the print data passed to the PrintToJournal( ) API **Author**, modify the print data associated with existing Journal Page workers
Receipt Printing	**Registry**, for character mappings for other vendors' printers **Author**, for CDI stores for printing (Appendix C, "Common Data Interface Stores")	**Registry**, for character mappings **Author**, for CDI stores for printing (Appendix C, "Common Data Interface Stores")
Statement Printing	**Registry,** for printers using XFS forms (Chapter 5, "Configuring Advance NDC and Support Applications")	**Registry,** for printers using XFS forms (Chapter 5, "Configuring Advance NDC and Support Applications")
NDC host protocol (VPI comms)	A new VPI communications protocol is supported by installing a new communications product that supports the VPI comms interface. You need to provide a method of configuring the protocol, for example, through a new Supervisor function	N / A

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Implementation Task	Create	Modify
UCDI Store	**Registry,** create a registry key if you need ease of configuration and persistent **Author**, create a UCDI store when you need information to be available across authored and non-authored processes (Chapter C, "Common Data Interface Stores") Create a sharable Integer Store for cross-authored-process sharing	N / A
Code Page	**For printing**, to select a different code page, modify the appropriate control codes defined in the K-screens in the *resrvd.def* file. These control codes are sent to the printer at various points in the application flow to initialize the printer (start of day, exiting Supervisor and so on) For temporary modification of the code page, embed the appropriate printer control codes in the print data either sent from the host (using transaction reply print data) or from printer-related workers in the application flow. To create your own character set character set mapping can be used to select existing characters in code pages **For screen display**, Advance NDC uses the concept of a font rather than a code page: embed the appropriate font selection control codes in your screen definitions or install a new True Type font. For operator screen display (on the front), embed the appropriate font selection control codes in the *resrvd.def* strings used to display the supervisor menus. (*Advance NDC Reference Manual*)	N / A

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Appendix A

Using Exits in Advance NDC

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Overview

This appendix gives additional guidance on designing and implementing Exits in Advance NDC. It includes the following:

- The various methods available to you for implementing Exits.
- A comparison of the methods to enable you to select from:
  - C Exits
  - Advance NDC Authored Exits
  - ActiveX Exits
- Design considerations for the portability of Authored Exits
- How multiple exit hooks can be used.

In addition, reference to additional documentation for each method is given. This documentation can be elsewhere in this Developer's Guide or in other Advance NDC publications.

For the knowledge and experience required to design and implement exits in Advance NDC, see the "What Experience Should I Have?" section in the Preface.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Methods of Implementing Exits

The term 'Exit' is used to describe any extra functionality or modules that are designed to interface with the standard released version of APTRA Advance NDC.

The following implementation methods are available to you:

- C Exits
- Authored Exits
- ActiveX Exits.

These implementation methods and their advantages / disadvantages are described in the following sections.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

C Exits

'C Exits' are Exits implemented as DLLs written in C or C++ source code. The 'C' in C Exits refers to the interface, which is a pure C interface provided by Advance NDC, offering external code access to the Advance NDC systems and data. For details of the interface, see Appendix C, “Common Data Interface Stores”.

The Application Core internally launches *aandcia.exe*, which manages the invocation of any C Exit extensions.

C Exits are supported in Advance NDC to offer a migration path from NDC+ installations. There are some differences between the NDC+ and Advance NDC versions of Exits support. For details, see “Differences in the Development Process” in Chapter 3, “Migrating Existing NDC+ Applications to Advance NDC”.

For information about accessing the CEN-XFS interface used for certain devices, refer to the white paper, *CEN-XFS Exits in Advance NDC*. A link to the white paper is available through the Advance NDC topic in the APTRA Documentation. (Select **Start | Programs | NCR APTRA | APTRA (TM) Documentation.**)

Advantages of C Exits

C Exits offer the following benefits:

- **Power**
  Being closer to the Windows XP operating system, C Exits have full use of the system resources through the Microsoft Win32 Application Programming Interface (API) or similar libraries. Therefore, C Exits can be more powerful than Authored Exits.
- **Portability**
  As C Exits are self-contained units, they can be migrated to new versions of Advance NDC.
- **Easy removal for debugging**
  You can edit the registry files to add or remove C Exits from the system to assist with debugging.

Disadvantages of C Exits

Consider the following before starting to write a C Exit:

- Writing C Exits can be time-consuming and demanding as skill in C or C++ is required, unlike the Author, which was developed for rapid application development without the need to learn a complex programming language.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

- For financial applications, the C interface with Advance NDC may be limiting. C Exits can communicate with Advance NDC only through the C interface. Therefore you have to allocate and free memory in your code and explicitly write updates to system Data (CDIs)
- The interface with the APTRA platform is limited. Your C Exit must use the CEN-XFS API. However, not all devices are available to the C Exit code.
- As C Exits are not in the Author worker environment, your code must manipulate SST devices without the help of the supporting workers.
- The System State is difficult to monitor. For example, device fitness or communications state may be an issue.

For details of device access, see “Device Access” on page 4-7.

Reasons for Using C Exits

C Exits are useful if speed of execution is important as C Exits give the best speed for large data processing operations.

C Exits can also be used where portability between Advance NDC systems is important.

For more information about migrating C Exits from NDC+ to Advance NDC, see “Migrating Existing NDC+ Exits” in Chapter 3, “Migrating Existing NDC+ Applications to Advance NDC”.

Implementing C Exits

Like NDC+, APTRA Advance NDC will call C Exit routines at various points in the logic flow if they are present.

You implement the C Exit as an exported C function within a DLL. You must declare the presence of a C Exit in a registry file to indicate it to the system. As a minimum, you must define in the registry file the DLL and the entry point function of your C Exit. Depending on where the C Exit is called, it will be referred to by a different name but the implementations of all the C Exit types are very similar.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

The following table shows which registry file and entry points to use for each Exit type:

Table A-1
Registry File and Entry Points for Exit Type

Exit Type (*Worker*)	Point of Call	Registry File
Exit State *(NDC Processing State Worker)*	State Execution	STCONT
Exit Hook *(Generic Exit Executor)*	Supervisor Entry Supervisor Exit Supervisor Stop Clear Fitness * Clear Device * Suspend * Initialise Exit Close State	MISCONT *See* Note:
Supervisor Exit *(Supervisor Exit Executor)*	Supervisor Function	SUPCTR

* Not supported by Advance NDC.

Note: Multiple exits are supported by the provision of a new *MISCMULT.DLL* file. To use *MISCMULT.DLL*, *see* the "Multiple Exit Hooks" section.

Implementation Design

The system C functions available to C Exits are limited compared to those available for an Authored Exit. The Exit DLL will have to deal with its own screen and keyboard handling and can only display on the Cardholder (NDC) screen using the DisplayScreen API. Supervisor screen display is difficult and not recommended as there is no API support.

Device handling must be through the APTRA platform's XFS interface, which will require detailed knowledge of the devices. In addition, you will need to write your own Timers if they are required.

Exit states normally return the next state to the system. If the Exit modifies any CDI data, then it is important to remember to store the new values back to the system at the end of Exit processing.

Basic Implementation Procedure

Complete the following steps for all the above Exits:

1 Implement the Exit as an exported C function within a DLL. This function is exported as an 'extern C' and will be called by Advance NDC.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

2 Include Advance NDC headers in the Exit DLL to access the system C functions. The headers and libraries required are provided as part of the Advance NDC development system.

   For more information about the differences between NDC+ and Advance NDC header files, see "Differences in the Development Process" in Chapter 3, "Migrating Existing NDC+ Applications to Advance NDC".

3 Edit the appropriate registry file to make the system aware of the presence of the Exit.

4 Copy the updated registry files and Exit DLLs to the destination machine.

   When the system is run, the Application Core application will load all the Exit routines to verify that their entries in the registry files are valid. The routines will then be called by the appropriate hook points as the system flow executes.

You can use the Tester and Assigner workers to call an external C function by first declaring the function and DLL in the *userfuns.scr* file located in the following folders:

- On the development PC
  <global> \ test \ support \ user
- On SSTs
  <global> \ final \ ssds \ dll

The *userfuns.scr* entry has a fixed format such as:

```
Register DLL "ssdsNDCFUNS" Function "NDCReleaseAllDevices" Returns
void
```

The format of the C source code is the same as that of the other Advance NDC C Exits as regards the calling convention used. The interface must be a C interface. For more information, refer to the *APTRA Advance ADE, Programmer's Guide*, Section 2 - External C Functions.

Note: Function names either in the *userfuns.scr* file or declared in the code must not exceed 64 characters.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Authored Exits

Exits from the state flow to an Authored state are achieved using an Exit Condition Tester worker.

Other Exits such as Supervisor entry, exit and menu can be implemented directly in the application using the Author.

Advantages of Authored Exits

The full range of Author workers is available to the Exit programmer and so all devices and the state of the system are available, with the following benefits:

- Screen displays are easily added.
- Existing workers can be reused by copying them from other areas of the application, thus reducing development time.
- Parts of the application can be shared, again reducing development time.
- System calls can be achieved by using VBScript within the Active Script Host worker, making the system drive, registry and so on available to the Exit.
- Some message customisation has been built into Advance NDC to allow the modification of the message interface. For details, see “Processing a New Message Class” in Chapter 8, “Enhancing the Application Core or Supervisor”.

Disadvantages of Authored Exits

Authored Exits are less portable than C Exits between different versions of Advance NDC implementations. When upgrading to a newer release of Advance NDC, or installing a Service Pack update for Advance NDC, time will have to be spent reapplying the customisations to the new release or update.

For details of device access, see “Device Access” on page 4-7.

Reasons for Using Authored Exits

Authored Exits are recommended every time over C Exits due to the availability of the Author development environment, and should be the first choice when developing an Exit for the following:

- New states, Supervisor functions, Supervisor exit and entry; start up, suspend and close state Exits.
- Any device interaction, particularly screen and keyboard.
- Web-enabled Exits, which must be implemented with Authored Exits.

Note: Authored Exits can call C Exits at any time, using the Evaluator class.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Implementing Exit States

As an example of implementing an Authored Exit, here we look at the implementation details of the Cheque Accept state.

The steps in implementing this example are:

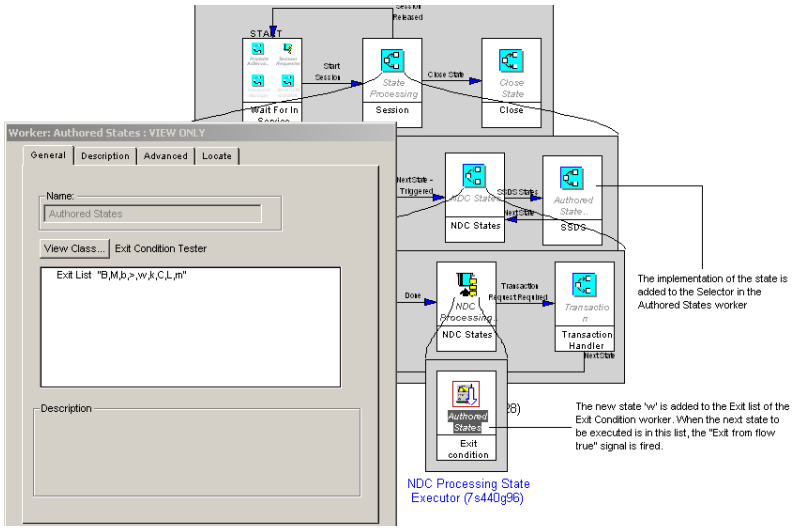
1. Define the state in the state table entry; the state ID chosen is 'w'.

2. Update STCONT to inform the system of the presence of the new state and the range of valid table entries. If the 'Advance NDC or Exit Support Flag' is E, the DLL name and Function name fields are filled with '-' as there are no external functions to call.

   For details of the format of the entries in STCONT, see “STCONT Format” in Chapter 3, “Migrating Existing NDC+ Applications to Advance NDC”.

3. Integrate the new state by adding its implementation to the selector in the Authored States worker, and adding the new state ID to the Exit list of the Exit Condition Tester worker, as shown in the following diagram.

Figure A-1
Integrating an Authored Exit State



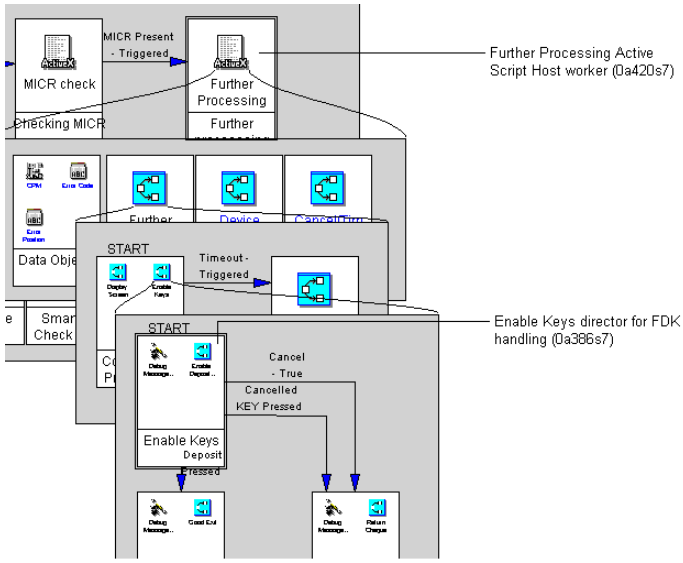
Within a state, all features of the Author can be accessed, including the workers for screen, keyboard and device handling. This means useful Authored flows can be copied from existing Authored states to speed up development. For example, FDK handling can be copied from:

```
Cheque Accept state> FurtherProcessing>Enable Keys
Director (0a368s7)
```

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

shown in the following diagram:

Figure A-2
Reusing an Authored Flow



The Customisation Layer and Application Core applications have been enhanced from Advance NDC 2.01 to enable modification of the Terminal Message. For details of this see Chapter 7, “Enhancing the Customisation Layer” and Chapter 8, “Enhancing the Application Core or Supervisor”.

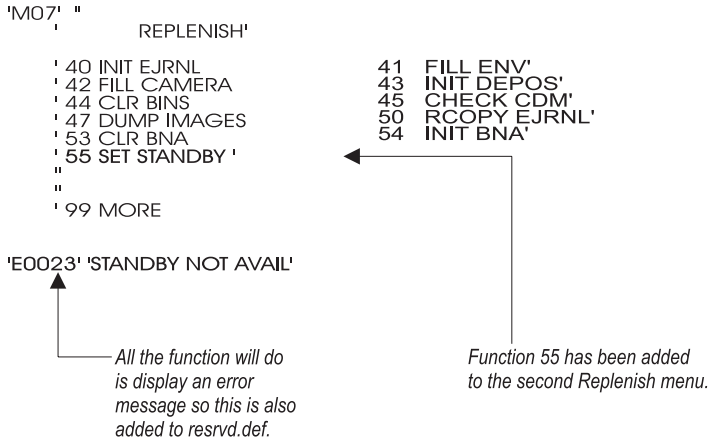
Implementing Supervisor Exits

New Supervisor functions are easily implemented using the Author. For example, a proposed feature may be to include a standby option on SSTs to save energy over periods of inactivity. A Supervisor function will be required to set a value for the timer that will signal the start of the standby mode. This can be implemented as follows:

1 Edit the *resrvd.def* file to include the new menu selection on the appropriate menu by adding text in the same format as existing entries. Bear in mind the space available on the operator panel display. In the following example, the added text is in bold. As the function will only display an error message, the error message is also added to the file.

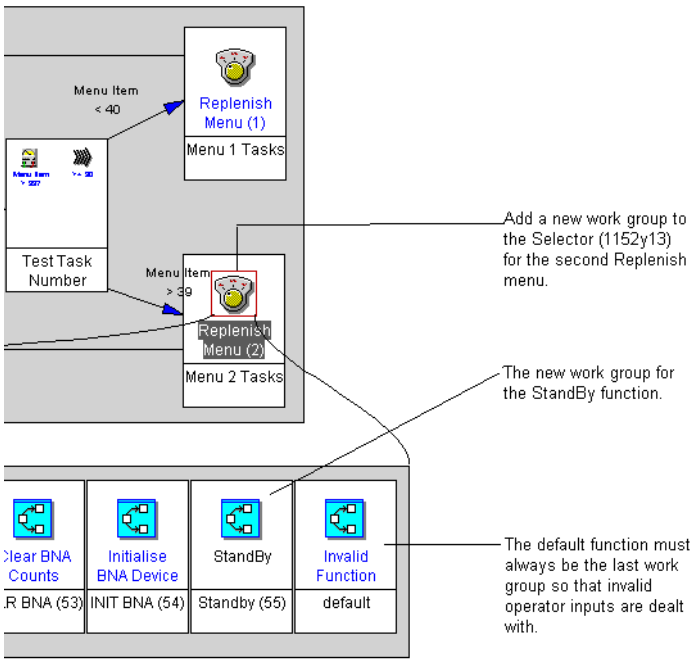
Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Figure A-3
Editing RESRVD.DEF



2 In the Application Core Author project, add the function to the Supervisor mode.

Figure A-4
Integrating a New Function in the Application Core Author Project



To reduce development time, you can copy the operator keyboard handling and error handling from existing Supervisor functionality.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

ActiveX Exits

The Active Script Host worker is a very powerful feature within Advance NDC. With this worker, the authored application can execute a script to manipulate workers in more efficient ways than if purely authored. In addition, the world of COM / ActiveX objects are made available to the author environment, providing unlimited resources to enhance a terminal application.

An Active Script Host worker can be used anywhere in the application. At the simplest level it can be used to call C++ Exits with a COM interface, its job being to choose which COM Exits to call, depending on the state of the application.

Debugging the Script

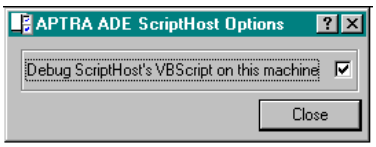
Debugging a script requires a debugging tool such as Microsoft Visual InterDev to be installed. Setting this option enables just-in-time (JIT) debugging, prompting you to launch the tool for each invocation of an Active Script Host. You can then step through the code and examine the contents of variables.

To activate script debugging, select **Run** on the Windows **Start** menu and enter

<drive:> \ <global> \ test \ xfs \ dll \ axscripthostoptions.exe

to display the following dialog window. Check the box as shown and close the window.

Figure A-5
Debugging Scripts Option



Advantages of ActiveX Exits

ActiveX Exits have the following advantages:

- The development of the Exit can be performed separately from the main application.
- The Exit can be updated without affecting the main application. As long as the globally unique identifier (GUID) is the same, updates to the Exit can be introduced on the development system or SST with minimum disruption.
- The Exit developer can choose any implementation language as long as the Exit uses a COM interface. For example, the implementation language can be APTRA Advance ADE, C++ / ATL / OLE, Visual Basic, or Java.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

- If the object supports the *IProvideClassInfo2Impl* interface, it may be accessed using an Automation Object worker.
- The COM CDI extensions will make Advance NDC system data available to the ActiveX Exit.

Disadvantages of ActiveX Exits

The use of COM requires a specific skill set that takes time and effort to acquire.

For details of device access, see “Device Access” on page 4-7.

Reasons for Using ActiveX Exits

The advantage of using an ActiveX Exit is that the Exit is a self-contained file, like a C Exit DLL, and therefore the main application need not be dependent on it.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Portability of Authored Exits

This section discusses design considerations if you plan to modify the APTRA Advance NDC Application Core and Customisation Layer applications. In addition, please read the relevant guidelines in documentation listed in Appendix E, “APTRA Author Documentation”.

The flexibility of authored applications, such as the Customisation Layer in Advance NDC, makes it easy to enhance Authored applications; however, this does not always result in well-designed, reusable software.

Reuse is not always considered a design issue when each version of the application is regarded as unique. However customisations will have to be reapplied when upgrading to a new release of Advance NDC or installing a Service Pack. Re-applying customisation can be time-consuming.

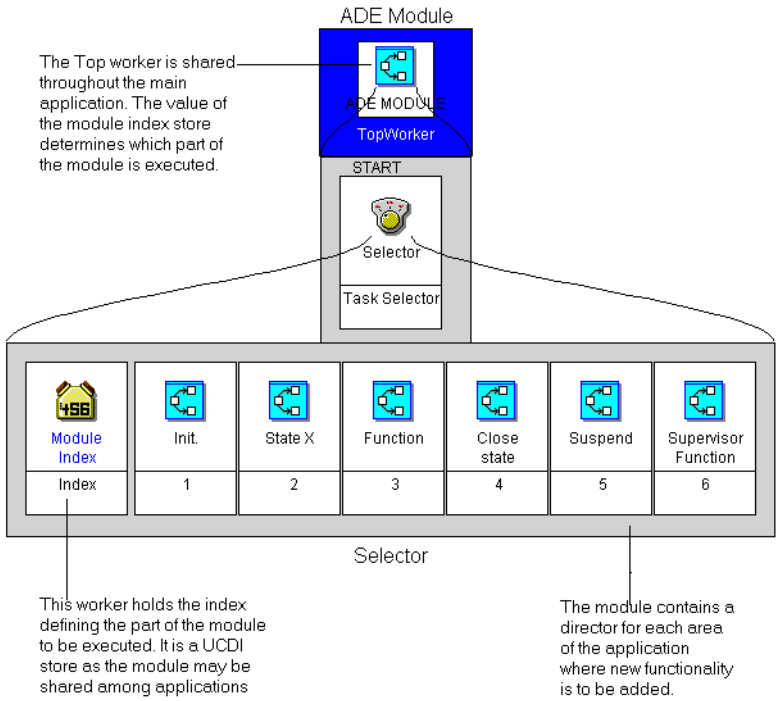
A modular approach to the construction of any enhancements enables such enhancements to be easily shared among unique applications. Modularity can be achieved by using the techniques described in the following sections.

Using Single Entry And Exit Points

This technique involves using a single worker as an entry / exit point to the module under development. The module is shared throughout the application where there is new functionality. A single flag is used to indicate to the module the task to be performed. The module can be authored to have a Top worker. The following example shows a Top worker being used.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

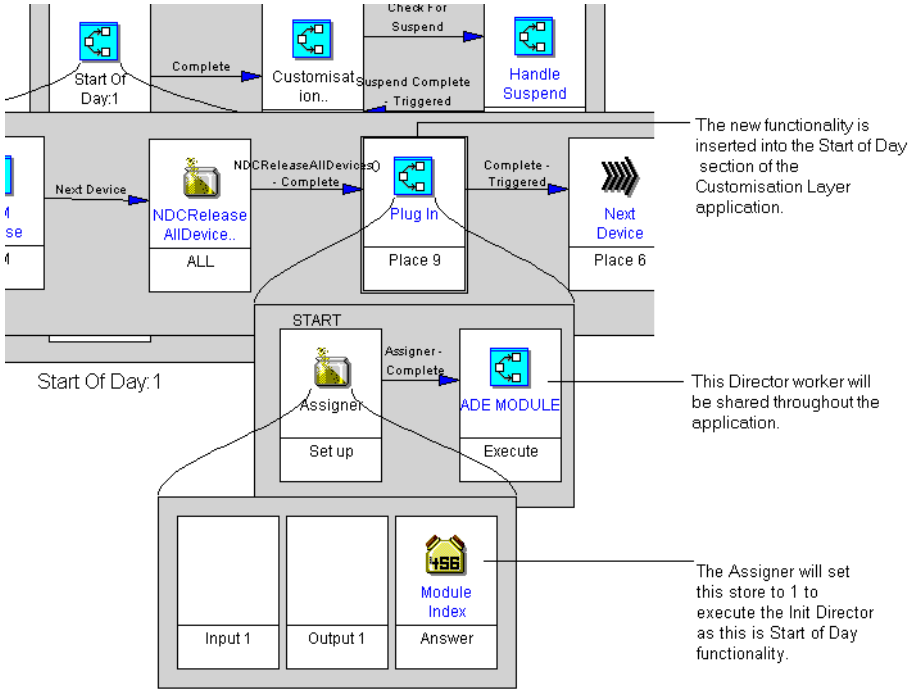
Figure A-6
Single Entry and Exit Point



In the main application, the Top worker will be shared throughout, and the index changed to reflect the required functionality. The following diagram shows the new module added to Start of Day.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Figure A-7
Adding the Module to the Main Application



It is important for only one signal to be produced by the new module to make it easier to add to existing applications. Values can be returned using the CDI or UCDI stores.

Avoiding Empty Work Groups

Authored applications, including those supplied with Advance NDC, are multi-threaded applications. Therefore, when authoring modules for portability, never introduce dead ends, that is, empty work groups. Always produce a signal to indicate that the module has finished processing. This will enable the calling application to wait for your module to complete before continuing.

Porting an Authored Exit

It is simple to port an Authored Exit if you use a single catalog, and record in a document the entry points and other useful instructions.

Using a Single Catalog

Using the above method only the Plug In director (shown in Figure A-7, “Adding the Module to the Main Application”) will be delivered within a catalog. This will then be replaced along with the Assigner so that ownership of these workers can be transferred to the Author ID of the new user.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

The only work that a new user needs to do is to add the Plug In director to the appropriate points in their application. The Assigner will also need to be changed at each point to ensure the correct part of the module is executed and all signals are re-assigned.

Documenting the Entry Points

The list of entry points for the new module can be described by using annotated screenshots like the ones used in this appendix. You can produce these by pressing the ***Alt+Print Scrn*** keys when the Author window is active to capture the screen. This places a screenshot on the Windows clipboard. You can then use an application like Microsoft Paint to paste, crop and annotate the image. If you use MS Word for the document, you can annotate the image in MS Word by using callouts.

Note: NCR recommends that you refer to workers by their Worker ID.

Reusing Authored Work

To reuse authored work, use the Replace function rather than the Module Copy function. Do this as follows:

1. Take a copy of the Application worker to your own catalog. Notice all of the workers in the application are now shared.

2. To change some functionality of the application, replace workers down to the level of the change. Any workers that produce signals will need to have their signals re-assigned.

When replacing workers, rename them as your own so that they are easily identified as replacements in the future.

Advantages of the Replace Function

Replacing workers is initially time-consuming as you must also re-assign the signals. The benefit is realised when there is a new release of the application.

If workers are replaced, the new application can be merged in and all the areas that have not been replaced will be updated with the new workers and functionality. You need only compare the areas that you have replaced in your application with the new version of the application and manually pick up changes.

Limitations of the Module Copy Function

For Advance NDC applications, avoid using the Module Copy function, as this changes every worker in the module from a worker issued by NCR Dundee to one belonging to you. This has the effect of issuing new worker IDs to all the workers.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Thus, when a new release of the Advance NDC application is made available, merging a module that has been copied will not be successful, and any updates will not be applied to your module.

Instead, selectively copy or share workers as required.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Multiple Exit Hooks

Multiple exit hooks can be run at the same time (if required) by using the *MISCMULT.DLL* file provided.

If you have already defined C Exits in your MISCONT registry file, or if you plan to use other Hook Exits with the 'EMV/CAM2 Exits for APTRA Advance NDC' product, you will need to use the new *MISCMULT.DLL* file and you must use release 1.xx of 'EMV/CAM2 Exits for APTRA Advance NDC'.

MISCONT Rule and MISCMULT.DLL Files

Each standard hook in MISCONT must call the required public hook function in the *MISCMULT.DLL* file.

MISCONT Rule (Definition) File

You must create the MISCONT file, containing entries of the format:

1<gs>miscmult<gs>SuperEntryHooks<fs>
2<gs>miscmult<gs>SuperExitHooks<fs>
3<gs>miscmult<gs>StopSuperHooks<fs>
8<gs>miscmult<gs>StartOfDayHooks<fs>
9<gs>miscmult<gs>CloseStateHooks<fs>

Each entry calls the corresponding function in the following MISCMULT.DLL file.

MISCMULT.DLL File

The corresponding SuperEntry and SuperExit hook functions in the MISCMULT.DLL file then call:

- for SupEntryName, the SENTCONT rule file
- for SupExitName, the SEXCONT rule file
- for SupStopName, the STSCONT rule file
- for InitialiseExitRoutine, the INITCONT rule file
- for CloseStateName, the CLSCONT rule file.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and/or distribution is strictly prohibited.

Definition Files

You will need to create the following rule (definition) files for their corresponding hook functions in MISCMULT.DLL:

Table A-2
Rule Files

Definition File Name	Advance NDC Hook Function Name	Point Of Use Value / Usage Attribute
SENTCONT	SupEntryName	1
SEXCONT	SupExitName	2
STSCONT	SupStopName	3
CFCONT	ClearFitnessName	4 - *See* Note:
CDCONT	ClearDeviceName	5 - *See* Note:
SUSCONT	NDCSuspendName	6 - *See* Note:
INITCONT	InitialiseExitRoutine	8
CLSCONT	CloseStateName	9

The Generic Exit Executor worker class is used to call the required hook function. The Generic Exit Executor is used as described in the *APTRA Advance NDC, Worker Class Help* (h10044.hlp).

Note: These values are not supported by Advance NDC. The Generic Exit Executor 'Usage' attribute 7 (ProcessHoCommMsgName) is also not supported in Advance NDC.

Each definition file must contain entries of a similar format, as follows:

```
HookDLLName<GS>FunctionName<FS>
2ndHookDLLName<GS>2ndFunctionName<FS>
3rdHookDLLName<GS>3rdFunctionName
```

where:

<GS> = Group Separator (1DH)
<FS> = Field Separator (1CH).

For example, if more than one Start Of Day hook is required, the MISCONT entry has the format:

```
8<GS>MISCMULT<GS>STARTOFDAYHOOKS<FS>
```

This causes the `StartOfDayHooks` function to be run, which will parse the INITCONT file.

If the INITCONT file contains the following entries:

```
CUSTOMEXIT<GS>CUSTOMSTARTOFDAY<FS>
CUSTOMEXIT1<GS>CUSTOM1STARTOFDAY<FS>
```

this causes the `CustomExit` and then the `CustomExit1` `Start Of Day` actions to be run.

Using this method, more than one action for each hook function can be defined.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Note: The definition files must be copied to your *custom* directory. This includes them in the aggregate when you export it from the Aggregate Builder Tool for installation on an SST.

For further details of MISCONT and the other files relevant to Exits, refer to the *NDC, Using NDC Exits* publication.

Error Handling

If the MISCMULT.DLL encounters any errors when attempting to load or call any of the routines specified in the new definition files, the normal Exit DLL load and call errors are logged to the journal and the hook is not called. The next hook in the definition file is then actioned as normal.

All the routines specified in the SEXCONT definition file must return the same destination mode. If not, the destination mode returned by the last Supervisor Exit Hook called is used.

If any of the new definition files are required but not present, no action is taken by *MISCMULT.DLL*.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Appendix B

Advance NDC Workers Supplied

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Overview

To enhance the provided Advance NDC application, you need to visually link workers together. Workers are the components in an authored application. They are designed for maximum re-use, and so reduce the effort and time required to develop the application.

NDC-specific workers are provided for the *CustomisationLayer.mpj*, *ApplicationCore.mpj* and *Supervisor.mpj* APTRA Author projects.

Additional Advance NDC worker classes (not listed here) are provided for inheritance purposes only. These worker classes are called abstract classes; you cannot create an instance of an abstract class.

For a detailed description of the workers, including attributes, signals, and work groups, refer to the on-line help in the APTRA Author.

Note: For details of worker support in Advance NDC, refer to the *APTRA Author, User's Guide*, Appendix B, "Advance NDC Worker Support".

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Customisation Layer Workers

The **Exit Condition Tester** signals Exit From Flow True, when a State Number/Type in its Exit List attribute matches the Common Integer Store's 'Current State Number' or 'Current State Type'.

The **NDC Close State Executor** is used to execute an NDC+ State Flow between the Close State and State Number 0. It offers a choice of two methods to remove the last NDC+ screen, before returning to the Author flow. It exits on a set of conditions checked for by an Exit Condition Tester worker.

The **NDC Data Collector** allows the front keyboard and front FDKs to be enabled for data collection, and data to be echoed to the screen during data entry.

The **NDC Echo** worker allows data to be displayed on the NDC screen, and stored as a string on completion of data entry.

The **NDC PIN Collector** allows the customer to enter a Personal Identification Number (PIN). It displays an NDC screen determined by the state download for the state type, and waits until a PIN is entered or a timeout occurs.

The **NDC Print Footer** worker is used in the Close state to print any remaining preprint information as a receipt footer and eject the receipt.

The **NDC Processing State Executor** is used to execute NDC+ States after a card has been entered, and signals when the next state to be executed is a Transaction Request or Close State. It exits on a set of conditions checked for by an Exit Condition Tester worker.

The **NDC Timeout State Executor** performs the same functionality as the NDC+ Timeout State. Screen C00 is displayed, giving the cardholder the option of selecting more time in the transaction, and an intermittent beep is sounded. Whichever key is pressed, the beep is stopped before signalling.

The **PIN Retries Updater** reads the integer data supplied by the worker in its Retry Count work group, and uses it to update the retry count data stores. This worker only updates the Retry Count and Card Track Data Stores; the Card Track Data is not written to the card.

The **Random PAD Character** worker supplies a random PAD character in the range hex '0' to 'F'.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and/or distribution is strictly prohibited.

The **Session Releaser**, together with the Session Requester worker, provides interfacing and synchronisation between the Customisation Layer and the Application Core. It enables the Application Core to communicate the mode has changed, preventing another cardholder session.

The **Session Requester**, together with the Session Releaser worker, provides interfacing and synchronisation between the Customisation Layer and the Application Core. It signals when the Application Core enters In Service mode, permitting the start of a new cardholder session.

The **XFS Front FDK** worker has the following two uses:

- If used within the NDC Data Collector worker, it provides the key codes to use as part of the data input process.
- If used concurrently with an NDC Data Collector, detects key presses based on the FDK Key Code attribute or Dynamic Key Code work group.

The **XFS Front Key** worker can be used to enable and detect any key or group of keys on the cardholder keyboard.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

NDC Core Workers

These workers may be used by one or more of the three APTRA Author projects.

The **Barcode Reader** worker is used to read barcode information. Once the barcode has been read by the device the worker stores the result and produces a success signal. If there is any problem with the device the worker produces a failed signal.

The **Barcode Reader Resource ID** worker is used with resource management classes to provide management for the Barcode Reader device. This worker class can be used in any work group that accepts Resource Identifier workers.

The **Cancel Collector** signals when the Cancel key is pressed, indicating that data collection is complete.

The **Card ATR Reader**, used for a DASH reader, gets the Answer to Reset from the chip on the card and stores it.

The **Cheque Acceptor** worker enables the CPM for cheque acceptance, reads the codeline on the cheque and scans an image of the front and / or back of the cheque.

The **Cheque Cature** worker captures cheques.

The **Cheque Ejector** worker ejects cheques.

The **Coin Dispenser** worker is used in coin dispensing operations from one or more coin hopper types.

The **Common Integer Store** worker stores a piece of integer data, enabling access to the data from within any authored application.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

This worker class can be used anywhere an Integer Giver or Integer Receiver can.

The **Common Real Store** worker stores a piece of real data, enabling access to the data from within any authored application. This worker class can be used anywhere a Real Giver or Real Receiver can.

The **Common String Store** worker stores a piece of string data, enabling access to the data from within any authored application. This worker class can be used anywhere a String Giver or String Receiver can.

The **Configuration Data Saver** worker processes data received in a Configuration Parameters Load, Enhanced Configuration Parameters Load, or MAC Field Selection Load message. It saves the configuration data, specified by the String Giver worker in its Config Data work group.

The **CPM Initialise** worker sends the Reset command to the CPM, attempting to clear the device fault and return the CPM to a known good state and available for use.

The **CPM Resource ID** worker is used to determine whether the CPM device is present. It also enables the fitness of the CPM device to be checked when used in the Fitness Getter worker.

The **CPM TI** (cheque processing module tamper indication) worker is used to detect the removal or insertion of a CPM bin.

The **Customisation Data Decomposer** worker validates and stores NDC+ customisation data, as specified by the String Giver in its Custom Data work group.

It can handle data from the following messages:

- State Table Load
- Screen / keyboard Load
- FIT Data Load
- Override Reserved Screen Configuration.

The **Customisation Data Initialiser** worker initialises the NDC+ Customisation Data to default values, or to previously downloaded and saved values, according to the Configuration ID. A signal is produced on completion of the customisation data initialisation.

The **Customisation Data Saver** worker saves the NDC+ Customisation Data, which includes the State Tables, Screen, Keyboard and FIT customisation data, to disk. It ensures the data is saved across any power failure, avoiding the need to download the data from Central on power-up.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

The **Debug Message Sender** worker allows application debug messages to be sent to a server program (DebugLog) where they can be displayed as the application is run.

The **Disk Size Getter** worker determines the size in bytes of the disk specified by its Drive attribute.

The **Extended Cash Stacker** worker stacks a specified number of notes from each cassette type in preparation for presenting them to the cardholder.

The **First Cash Handler Resource ID** worker is a Cash Handler Resource ID for use with a dual cash handler and returns information about the first dispenser.

A **FIT Searcher** worker is used to perform an NDC+ FIT match. This worker searches the Financial Institution Table (FIT) for a valid match, using information in tracks 1, 2 and 3. The search terminates on the first match.

The **Generic Exit Executor** worker loads and executes the function specified in the MISCONT registry file. The Completed signal is produced once this has been done. If the MISCONT registry file does not exist, this worker does nothing, and the Completed signal is immediately produced.

The **Integer Array Element** worker provides access to one element of a CDI array.

The **Journal Security Trace Generator** worker sequentially generates and returns a new security trace number which is required when printing status messages to the journal printer. This worker is used within Journal Page workers, when Security Trace Messages require printing within Advance NDC.

The **Linked FIT Loader** worker retrieves the next FIT Record from the downloaded table of FIT Records. The selected FIT Record is compared with the PIDDX, PFIID, PCKLN and PINDX fields to determine whether it is a linked FIT Record.

The **NDC Command Reject Message Builder** worker builds a Command Reject or Specific Command Reject Solicited Status Message, in NDC+ defined format. The message is stored by a String Receiver in the Message work group, before a Message Built signal is produced. Send using an NDC Message Sender worker.

The **NDC Comms Connection ID** worker uses the APTRA Comms Connection Manager (CCM) for communication with Central. By default, it uses the connection name 'TPA Connection', unless this is overwritten by setting the Name attribute. There must be only one instance of this worker for each connection.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

The **NDC Delete RSA Key** worker extracts the key data and signature from the Key Data work group. The data is decoded using the Base 94 encoding method and used to delete the RSA public key and signature in the encryptor.

The **NDC Device Status Message Handler** worker determines which Device Fault Status messages need to be sent to Central, and then builds and sends them.

This worker provides the option to send both Unsolicited and Solicited messages, or only Unsolicited messages.

The **NDC EPP Delete Key** worker is used with the EPP to delete a specified key and update its status in the encryptor.

The **NDC EPP Initialiser** worker is used only at Start Of Day to initialise the encryptor in the EPP (Encrypting PIN Pad). It checks the state of the Encryptor key spaces, and if required, initialises them.

The **NDC EPP Key Mode Initialiser** worker is used with the EPP after a key mode change has been confirmed, using the supervisor Access menu function '25 KEY ENTRY'. This worker determines whether the requested EPP key mode is different from the current mode held in the encryptor. If necessary, the key spaces are initialised for the new mode.

The **NDC EPP KVV Evaluator** worker is used with the EPP to generate the KVV (Key Verification Value) that confirms the correct loading of a specified encryption key.

The **NDC Export Encryptor Data** worker checks the Data to Export attribute to determine the data to retrieve from the encryptor (serial number, public key or capabilities / status) and sends the command to the encryptor. The retrieved data is stored in the Results work group (serial number, public key or capabilities) and the Signature work group (signature or status).

The **NDC Field** worker is used in the NDC Field work group of an NDC Transaction Handler worker, to signify the inclusion of a specific optional field in a Transaction Request message. The NDC Field ID attribute specifies the ID of the represented component field.

The **NDC Import RSA Key** worker loads any type of RSA public key (Host Public Key or Host Root Public Key) into the encryptor in the Encrypting PIN Pad (EPP). The key data and signature are obtained from the worker in its Key Data work group, The data is decoded using the Base 94 encoding method, and then used to load the RSA public key and signature into the encryptor.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

The **NDC Message Receiver** worker receives a message from the Virtual Message Buffer, or comms connection via the Virtual Controller. The message is stored in a String Receiver in its Mailbox work group. Since any virtual reply is stored in the Virtual Message Buffer, it is used with an NDC Message Sender.

The **NDC Message Sender** worker sends a message to the Central application or Virtual Controller. The message is specified and stored using a String Giver worker in its Parcel work group. Since any virtual reply is stored in the Virtual Message Buffer, this worker must be used with the NDC Message Receiver worker.

The **NDC Print Buffers** worker processes and prints the data received in the last Transaction Reply message, to the printer(s) specified by its Print Flag attribute. The worker signals Complete when printing of the required Print Buffers is complete.

The **NDC Print Coupon** worker sends a coupon image file to the printer, using the specified file name and XFS form.

The **NDC Print Immediate Handler** worker executes a Print Immediate message, supplied by the String Receiver in the Message work group. If a Transaction Reply message is successfully received and validated in the Tr Reply Function work group, the relevant Function is selected by its ID, and executed.

The **NDC Printer Initialiser** worker sends printer configuration data to the associated printers. It signals Successful once the data has been sent.

The **NDC Ready 9/B Message Builder** worker builds either a Ready 9 or Ready B Solicited Status Message. The default message type is Ready 9; a Ready B message is only used within Advance NDC in response to a Transaction Reply Message, if the relevant configuration option has been set.

The **NDC RSA Import DES Key** worker loads one of the DES keys (A Key, B Key or V Key) into the encryptor in the encrypting PIN pad (EPP). The A, B or V Key to be loaded is specified in the Destination work group, and the relevant key data and signature is contained in the Key Data work group.

The **NDC Screen Displayer** worker enables the re-use and display of previously downloaded NDC+ screen definitions, in any authored part of your Advance NDC application. You can specify customised screens using the Screen Number work group, or reserved screens using the Reserved Screen attribute.

The **NDC Terminal State Message Builder** worker is used to build a Terminal State Solicited Status Message, and store it using a String

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Receiver in its Message work group. The Status Information work group uses a String Giver to hold data for inclusion in a Terminal State Solicited Status Message. Send using an NDC Message Sender worker.

The **NDC Transaction Handler** worker enables execution of the complete Transaction Request / Reply sequence. It can be used as an alternative to the Transaction Request State worker.

The **Note Definitions Builder** worker is used to build a definition of the notes which are to be accepted by the SST, and store it.

The **Receipt Control** worker captures the receipt currently at the exit of the receipt printer.

The **Register Trace Generator** worker configures the Journal Configurator worker to generate a security trace header each time a message is printed during a journal print backup operation.

The **Resource Initialiser** worker sends a reset command to SST devices indicating a reset is required to recover from an error situation.

The **Resource Status Checker** worker reports the status of all devices in the Resources to Report work group. It produces a multi-line customisable report, using the Fault Display reserved screens. The status always reports error status messages and can be configured to report good status messages.

The **RSA Key** worker uses the Key Name attribute to obtain the key name or the signature key name of the RSA key to be imported or deleted.

The **Second Cash Handler Resource ID** worker is a Cash Handler Resource ID for use with a dual cash handler and returns information about the second dispenser.

The **Supervisor Exit Executor** worker is used after each menu / function selection, to implement NDC+ Supervisor functions in the Supervisor Exit. If the registry file specifies that a defined supervisor function is supported by an exit, an Executed signal is produced; otherwise Not Executed is signalled.

The **TCP/IP Comms Connection ID** worker is no longer supported, but may remain referenced within any application, in the NDC Comms Connection ID worker.

The **Tr Reply Function** worker executes a specific Transaction Reply (for example, Card Before Cash). It is used within the Tr

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Reply Function work group of an NDC Transaction Handler or Transaction Request State worker.

The **Track 1 Decomposer** worker should only be used if you require to decompose Track 1 data, and replace the provided NDC+ Card Read State with your own card entry, read and FIT match. The decomposed data is stored in workers in its work groups in the appropriate format. Track 1 sentinels are ignored.

The **Transaction Request State** worker enables execution of the Transaction Request State. It builds messages using the State Table parameters for current State Number and Configuration Parameters. It ensures that Unsolicited and Solicited Device Fault Status messages are sent in the correct order.

The **Variable Graphics Paragraph** worker specifies a graphics file to the printer coordinator.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Application Core Workers

The **Data Encoding Converter** worker prepares key data in an Encryption Key Change message to put into a Key Exchange worker, or a VISA key table. It takes the data in the Data In work group, converts it using the Input Encoding and Output Encoding attributes, and stores the result in the Data Out work group. If the data conversion is unsuccessful, the Error Code work group is updated with a value indicating the reason for failure.

The **NDC Application Terminator** worker generates an mError. It does not complete due to the mError terminating the application. The Error Number work group contains an integer store which specifies the type of mError.

The **NDC Configuration Info Builder** worker attempts to obtain the terminal's configuration information (hardware configuration, fitness, supplies status and sensor states), and store it using the String Receiver worker in its Status Information work group. This information is used to build the Terminal State response message for a 'Send Configuration Information' command.

The **NDC Counters Builder** worker obtains and stores the terminal's media count information using a String Receiver worker in its Status Information work group.

This information can include counts of currency, coins, documents, envelopes, camera film, captured cards and number of transactions. This information is used to build the Terminal State response message for a 'Send Supply Counters' command.

The **NDC DES Loader** worker provides the ability to load secret keys from clear data, which is needed to emulate the NDC+ supervisor functions Write A, B and V keys. The worker places the clear data supplied by the String Giver in its Data work group into the Programmer/Secret Key or Initialisation Vector in the Destination work group, packing the entered key according to whether it is single or double length.

The **NDC Dialup** worker controls access to starting and stopping sessions and transactions for dialup communications

The **NDC Error Log Builder** worker obtains error log information for the log group specified in its Log Group work group, and stores it using a String Receiver worker in its Status Information work group. This information is used to build the Terminal State response message for a 'Send Error Log' command.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and/or distribution is strictly prohibited.

Note: Advance NDC 3.0 returns a preconfigured default date and zero values.

The **NDC Fitness Data Builder** worker obtains fitness information for the terminal devices and stores it using a String Receiver worker in its Status Information work group. This information is used to build the response message for a 'Send Fitness Data' command.

The **NDC Hardware Config Builder** worker obtains terminal hardware configuration information and stores it using a String Receiver worker in its Status Information work group. This information is used to build the Terminal State response message for a 'Send Hardware Configuration Data' command.

The **NDC Message Validator** worker validates the message format of data held in its Message work group, which is typically a complete NDC+ message from the host.

The process of validation depends on whether the message is a Terminal Command, Configuration Data Load, Transaction Reply or Virtual Controller message.

The **Secure Encryption Key Collector** collects the encryption key data securely from the keyboard and returns the KVV value for the entered key.

This allows manual entry of a full-length, symmetric encryption key part directly into the PINpad without being exposed outside the PINpad.

The **NDC Sensors Status Builder** worker obtains information on the state of terminal sensors and tamper indicating devices, and stores it in a String Receiver in its Status Information work group. This information is used to build the Terminal State response message for a 'Send Tamper and Sensor Status Data' command.

The **NDC Supplies Data Builder** worker obtains the terminal's media supplies information, and stores it using the String Receiver worker in its Status Information work group. This information is used to build the Terminal State response message for a 'Send Supplies Data' command.

The **NDC Tally Builder** worker obtains tally information for the device specified in its Device work group, and stores it using the String Receiver worker in its Status Information work group. This information is used to build the Terminal State response message for a 'Send Tallies' command.

Note: Advance NDC 3.0 returns a preconfigured date and zero values.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

The **Reserved Screen Retriever** worker returns the specified NDC Reserved Screen data, taken from the RESRVD.DEF file.

You would typically use this worker in Supervisor Mode to retrieve data for display or printing.

The **Supplies Data Formatter** worker builds a string containing a description of any replenishable Self-Service Terminal devices currently requiring attention.

It is used only during execution of the 'Display Supplies and 'Print Supplies' Supervisor functions.

If all the device supplies are good, this worker returns a string indicating 'ALL SUPPLIES GOOD'.

The **UPS** worker is used for an uninterruptible power supply (UPS) device, enabling the SST to complete a transaction on battery power if the mains supply fails before going out of service, or shut down in a controlled manner. When activated with a state, this worker generates a detected signal if the state matches the actual UPS device state.

The **XML Config File Loader** worker is used to load the XML configuration file. It takes the input data from the Config Data work group, which is input to the XML configuration file. When the XML configuration file has been loaded successfully the worker produces a success signal. If there is any problem loading the XML configuration file the worker produces a failed signal.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Appendix C

Common Data Interface Stores

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Overview

As discussed in Chapter 1, “Introducing Advance NDC”, under the heading “Common Data Interface” on page 1-10, the Common Data Interface (CDI) and User-defined Common Data Interface (UCDI) consist of stores representing all of the data used throughout the Advance NDC application. There are several hundred CDI store workers, one for each piece of data.

The CDI stores are grouped into the following “CDI - <name>” catalogs:

- CDI - Barcode
- CDI - BNA Accepted Denomination Counts
- CDI - BNA Active Banknotes
- CDI - BNA Denomination Configuration
- CDI - BNA Deposited Denomination Counts
- CDI - BNA MISC
- CDI - Buffers
- CDI - Coin Counters
- CDI - CPM
- CDI - CPM MISC
- CDI - Dialup
- CDI - EJ Upload
- CDI - EMV
- CDI - Encryptor Variant
- CDI - Error Processing
- CDI - Exit Migration
- CDI - FIT Data
- CDI - Key Entry Mode
- CDI - Misc Counters
- CDI - Note Counters
- CDI - Option Digits
- CDI - Printing
- CDI - Screen Display
- CDI - Security
- CDI - State Information
- CDI - Supervisor
- CDI - Terminal Configuration
- CDI - Timers
- CDI - Transaction Processing Flags.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Note: There is also a CDI - Document Processing catalog. This catalog contains data related to the Document Processing Module (DPM), which is unsupported in Advance NDC.

The following sections give brief descriptions of all the CDI stores and UCDI stores provided with Advance NDC.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Barcode

Table C-1
Barcode

CDI Store Worker	Description
Barcode Buffer	This holds the barcode that has been read
Barcode Format	This holds the format of the barcode

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

BNA Accepted Denomination Counts

Table C-2
BNA Accepted Denomination Counts

CDI Store Worker	Description
Accept Count 1	This is used to hold the number of notes accepted for denomination type 1.
Accept Count 2	This is used to hold the number of notes accepted for denomination type 2.
...	...
Accept Count 50	This is used to hold the number of notes accepted for denomination type 50.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

BNA Active Banknotes

Table C-3
BNA Active Banknotes

CDI Store Worker	Description
Note 1 Active	This specifies whether denomination type 1 is active.
Note 2 Active	This specifies whether denomination type 2 is active.
...	...
Note 50 Active	This specifies whether denomination type 50 is active.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

BNA Denomination Configuration

Table C-4
BNA Denomination Configuration

CDI Store Worker	Description
DenomConfig Type 1	This holds the configuration information for denomination type 1.
DenomConfig Type 2	This holds the configuration information for denomination type 2.
...	...
DenomConfig Type 50	This holds the configuration information for denomination type 50.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

BNA Deposited Denomination Counts

Table C-5
BNA Deposited Denomination Counts

CDI Store Worker	Description
Denomination Type 1	This holds the deposit count for denomination type 1.
Denomination Type 2	This holds the deposit count for denomination type 2.
...	...
Denomination Type 50	This holds the deposit count for denomination type 50.
Total Notes to Escrow	This holds the total number of notes moved to the escrow.
Total Notes Rejected	This holds the total number of notes rejected by the BNA.
Total Notes Refunded	This holds the total number of notes refunded to the cardholder.
Total Notes Encashed	This holds the total number of encashed notes.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

BNA MISC

Table C-6
BNA Misc

CDI Store Worker	Description
BNA Denoms Clrd Date & Time	This holds the date and time the BNA counters were last cleared.
BNA Fitness Status	This holds the fitness of the BNA device.
BNA Maintenance Data	For BNA maintenance, this holds the diagnostics information returned from Self-Service Support in an error / warning condition.
BNA Option	This specifies whether BNA journalling is performed or not.
BNA Supplies Status	This holds the status of the BNA supplies.
Last Deposit Direction	This holds the direction of the last transaction completed by the BNA; either deposit or refund.
Option Array 45	This option digit specifies whether the BNA Transaction Counts are to be included in Field R (Last Transaction Status field) of the Transaction Request message sent to Central; sets the retract option, the number of notes to be accepted (up to 90 or more than 90), and whether to use the extended message format.
Programmed Currency Count	This holds the total number of denomination types (currencies) configured for the BNA device.
Solicited Status Message Sent Flag	This communicates to the Transaction Handler that a Solicited Status message has been sent to Central.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Buffers

Table C-7
Buffers

CDI Store Worker	Description
Decimal Character Code	This is set to '.' in the Card Read initialisation. It is possible to be set by the Amount Entry and the Enhanced Amount Entry. It is covered under the International Currency in screen handling.
FDK Keycode Buffer	This holds the FDK Keycode buffer, and is used in several of the State Types.
New PIN Buffer 1 Present Flag	This is used in the Transaction Request / Reply. If 'av' is required, then add in the Data Id 'U'. If this flag is TRUE, then include data from the Customer Selectable PIN Buffer 1 CDI in the Transaction Request message.
New PIN Buffer 2 Present Flag	This is used in the Transaction Request / Reply. If 'aw' is required, then add in the Data Id 'V'. If this flag is TRUE, then include data from the Customer Selectable PIN Buffer 2 CDI in the Transaction Request message as well.
Current Screen Number	This holds the Transaction Reply field 'm', which is the screen to be displayed during the Function Execution. Set to 0 when the Transaction Reply is received, then set to the correct value if field 'm' is included. If the screen number is non-zero, the screen will be displayed during the Function Execution.
Buffer B Decimal Point	This holds the decimal point position in the B Buffer and is set to 2 in the Card Read initialisation. It is used in the Amount Entry State and Amount Check State.
Buffer C Decimal Point	This holds the decimal point position in the C Buffer and is set to 2 in the Card Read initialisation. It is used in the Amount Entry State and Amount Check State.
Amount Buffer Decimal Point	This holds the decimal point position in the Amount Buffer and is set to 2 in the Card Read initialisation. It is used in the Amount Entry State and Amount Check State.
Amount Buffer	This holds data for the Transaction Request field 'k', which is the Amount Buffer.
Buffer A	This holds data for the Transaction Request field 'l', which is Buffer A.
Buffer B	This holds data for the Transaction Request field 'm', which is Buffer B.
Buffer C	This holds data for the Transaction Request field 'n', which is Buffer C.
Buffer f	This holds data for the Transaction Request fields 'ce1' to 'ce<*n+1*>', which is Buffer f. This is used only with a coin dispenser using more than four hopper types.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

CDI Store Worker	Description
Operation Code Buffer	This holds data for the Transaction Request field ‘j’.
Customer Selectable PIN Buffer 1	This holds data for the Transaction Request field ‘av2’. It is the first entry of a new PIN when doing a PIN change.
Customer Selectable PIN Buffer 2	This holds data for the Transaction Request field ‘aw2’. It is the second entry of a new PIN when doing a PIN change.
Card Track 1 Data	This holds data in the Transaction Request field ‘p’ and Transaction Reply field ‘ak2’, from or for Track 1 of a card.
Card Track 2 Data	This holds data in the Transaction Request field ‘h’ and Transaction Reply field ‘al2’, from or for Track 2 of a card.
Card Track 3 Data	This holds data in the Transaction Request field ‘i’ and Transaction Reply field ‘x’, from or for Track 3 of a card.
Virtual Message Buffer	This is used to store a message reply created by the virtual controller, rather than passing it to Central and waiting for a reply.
Terminal Status Buffer	This is used to store the status information field of a terminal status message, prior to the assembly of the message.
Device Fault Status Buffer	This is used to store status fields of a device fault message, prior to the assembly of the message.
Transaction Request Field Array	This is used to hold data written to it by the transaction request builder, after it has calculated which fields are to be included in the message. It is also used to add the MAC to the message once the virtual controller has completed processing.
Amount Buffer Length	If it is set to TRUE the Amount Buffer length is 12 instead of 8. This CDI holds the value of the Configuration Parameter or Enhanced Configuration Parameter (Part of Option 1).
Amount Read CAV Amount Entry Buffer CIM Verify Buffer CIM Verify Code	These are not used in Advance NDC.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Coin Counters

Table C-8
Coin Counters

CDI Store Worker	Description
Coins Remaining Hopper *x* Count	This holds the remaining coin counts for hopper type *x* An instance of this worker must exist for each hopper type
Coins Remaining Hopper *x* Backup Count	This holds the backup for the remaining coin counts for hopper type *x* An instance of this worker must exist for each hopper type
Coins Dispensed Hopper *x* Count	This holds the dispensed coin counts for hopper type *x* An instance of this worker must exist for each hopper type
Coins Dispensed Hopper *x* Backup Count	This holds the backup for the dispensed coin counts for hopper type *x* An instance of this worker must exist for each hopper type
Coins Last Dispensed Hopper *x* Count	This holds the number of coins dispensed in the last dispense operation from hopper type *x* An instance of this worker must exist for each hopper type
Standard Coin Count Hopper *x*	This holds the standard number of coins to be used for hopper type *x* during replenishment using the STD COINS Supervisor option An instance of this worker must exist for each hopper type
Coin Count Last Cleared Date	This holds the timestamp of the last coin counters clearing operation using the CLR COINS Supervisor option

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

CPM

Table C-9
CPM

CDI Store Worker	Description
Bills in Escrow Count	This holds the number of bills in the escrow.
Cheque Present	This indicates whether a cheque is present at the CPM
CPM Buffer a	This is used to store the codeline unique to each cheque, so that it can be used for identification purposes within the Transaction Request.
CPM Cheque Destination	This specifies the pocket in which to deposit/capture a cheque.
CPM Configuration	This is used to store the configuration of the CPM device at Start of Day.
CPM Endorse Flag	This holds the flag used to determine whether or not a cheque has been endorsed, so that the position of the cheque can be reported should an error condition occur.
CPM Pocket 1 Count	This holds the number of cheques deposited into Pocket 1.
CPM Pocket 2 Count	This holds the number of cheques deposited into Pocket 2.
CPM Pocket 3 Count	This holds the number of cheques deposited into Pocket 3.
CPM Print Data	This holds the data to print on the cheque, as specified by the Transaction Reply.
CPM Print on Reverse Flag	This holds the flag used to determine whether or not to print on the reverse of the cheque.
CPM Stamp Flag	This holds the flag used to determine whether to stamp the cheque once it has been deposited, as specified by the Transaction Reply.
MICR Detected	This holds the flag used to determine if the codeline from the cheque has been detected.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and/or distribution is strictly prohibited.

CPM MISC

Table C-10
CPM Misc

CDI Store Worker	Description
CPM Fitness Status	This holds the fitness of the CPM device.
CPM Pocket Count	This holds the total number of pockets associated with the CPM device.
CPM Pocket 1 Clr Date & Time	This holds the date and time of the last clearing action for Pocket 1.
CPM Pocket 2 Clr Date & Time	This holds the date and time of the last clearing action for Pocket 2.
CPM Pocket 3 Clr Date & Time	This holds the date and time of the last clearing action for Pocket 3.
CPM Pocket 1 Count	This holds the number of cheques deposited into Pocket 1.
CPM Pocket 2 Count	This holds the number of cheques deposited into Pocket 2.
CPM Pocket 3 Count	This holds the number of cheques deposited into Pocket 3.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Dialup

Table C-11
Dialup CDI Store Workers

CDI Store Worker	Description
Dial Up Enable	Flag to indicate whether the environment is a dialup environment
Dial Up Activity Timer	Number of 1-second units the application will wait before redialing and sending an I'm Alive message, after the connection has been dropped. The range is 0 - 25,500 seconds.
Dial Up Error Redial Timer	Number of 1-second units to wait before attempting to redial after a connection failure, a host timeout, or a failure to send the I'm Alive message or the Power-Up message. The range is range 0 - 2,500.
Dial Up Host Msg Pending	Flag indicating whether the host has messages to send, as follows: 1: host messages pending 0: no host messages pending
Dial Up Host Header Enabled	Flag indicating whether there will be a header to messages from the host, as follows: 1: host header expected 0: no host header expected
Dial Up Message Suppression	Flag to suppress SST to Central messages for Alarms and Supervisor key presses; journal update messages will still be sent.
Dial Up Msg Completion Option	Flag to indicate whether to send a transaction complete message to the host, as follows: 1: Message Type indicates that the SST *will* send a Transaction Completion message to the host. 0: Message Type indicates that the SST will *not* send a Transaction Completion message to the host.
Dial Up CP Message Timer	Number of 1-second units the application will wait for an additional message after sending a CP (or an EP) message to the host. The range is range 0 - 255.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

CDI Store Worker	Description
Dial Up Transaction Completion	Flag to indicate how to handle the connection and what message to send, as follows: 1: the SST remains connected after a Transaction Request (TREQ), will wait for a Transaction Response (TREP) and subsequently send a Ready message with CP. All message headers will have RQ, CP or CC. If the Message after CP Option is set to 1 then the application will wait for the time defined by the CP Message Timer before terminating the connection after sending the Ready message with CP. 0: the SST terminates the connection at TREP. All message headers will have EQ, EP or EC. If the Message after CP Option is set to 1, then the application will wait for the time defined by the CP Message Timer before terminating the connection. This option affects only the Ready message sent in response to a Transaction Reply command; All other Ready messages will still be sent even if this option set to 0.
Dial Up Last Msg Request	Set to 1 if the last message sent from Central was a transaction reply.
Dial Up Disconnect Received	Set to 1 when a disconnect command is received from Central.
Dial Up T Reply Received	Set to 1 when a transaction reply is received.
Dial Up Bin	A 7-character field containing information about communications access routing; used in the dialup header sent to Central.
Dial Up Terminal ID	A 6-character field containing information supplied by the host and used in the dialup header sent to Central.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

EJ Upload

Table C-12
EJ Upload

CDI Store Worker	Description
EJLastCharThisBlock	Holds the last character position of the current EJ data block sent to Central.
EJLastCharRecvd	Holds the last character position of the EJ data received by Central.
EJLastCharPrevBlock	Holds the last character of the previous data block received by Central. Binary zeroes are replaced with a question mark (?).
PendingEJFlag	Holds a flag indicating if there are pending EJ commands to be processed.
EJAckTimer	Holds a timer containing the timeout period allowed for Central to acknowledge an EJ Upload Message sent from the terminal.
EJRetryLeft	Holds the number of retries left before the terminal will automatically disable EJ Upload, when Central does not acknowledge an EJ data block within the timeout period.
EJBlockSize	Holds the size of (number of characters in) the EJ data block sent to Central.
EJRetryThreshold	Holds the maximum (threshold) number of retry attempts allowed before the terminal will automatically disable EJ Upload, when Central does not acknowledge an EJ data block within the timeout period.
EJUploadMsg	Holds the actual EJ Upload message sent to Central.
EJCommandCode	Holds the command code of the current EJ command being processed by the terminal.
PendingEJCommand	Holds the command code of any pending EJ commands waiting to be processed by the terminal.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

EMV

Table C-13
EMV

CDI Store Worker	Description
Currency Mapping Table	This holds the information used to define the currency held in the currency cassettes, such as currency type, cassette type and denomination.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Encryptor Variant

Table C-14
Encryptor Variant

CDI Store Worker	Description
EncryptorVariant	This holds the type of encryptor variant configured for the terminal: -1 = UNKNOWN 0 = NBS Service Variant 1 = BAPE Service Variant 6 = EPP Service Variant The default is -1.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Error Processing

Table C-15
Error Processing

CDI Store Worker	Description
Reject Status Value and Qualifier	This holds the Reject Status Value and Qualifier used in a Specific Command Reject message, for example A01.
CDI Error Number	This holds the Error Number to be reported. It is used internally within Advance NDC and is not intended for customer use.
CDI Error Message	This holds the Error Message to be reported. It is used internally within Advance NDC and is not intended for customer use.
CDI Error Parameter 1	This holds the variable String to be reported in the Error Message. It is used internally within Advance NDC and is not intended for customer use.
CDI Error Parameter 2	This holds the Variable String to be reported in the Error Message. It is used internally within Advance NDC and is not intended for customer use.
CDI Error Parameter 3	This holds the Variable String to be reported in the Error Message. It is used internally within Advance NDC and is not intended for customer use.
CDI Error Parameter 4	This holds the Variable String to be reported in the Error Message. It is used internally within Advance NDC and is not intended for customer use.
CDI Error Parameter 5	This holds the Variable String to be reported in the Error Message. It is used internally within Advance NDC and is not intended for customer use.
CDI Error Parameter 6	This holds the Variable String to be reported in the Error Message. It is used internally within Advance NDC and is not intended for customer use.
CDI Error Parameter 7	This holds the Variable String to be reported in the Error Message. It is used internally within Advance NDC and is not intended for customer use.
CDI Error Parameter 8	This holds the Variable String to be reported in the Error Message. It is used internally within Advance NDC and is not intended for customer use.
CDI Error Parameter 9	This holds the Variable String to be reported in the Error Message. It is used internally within Advance NDC and is not intended for customer use.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Exit Migration

The workers in this catalog contain data exposed in NDC+ but either not supported or is represented differently in Advance NDC. You should not need to use these workers.

Table C-16
Exit Migration

CDI Store Worker	Description
Option Digit 0	This holds the first Native Mode Option byte entered in the MSG MODE Configure Supervisor function.
Option Digit 1	This holds the 2ndNative Mode Option byte entered in the MSG MODE Configure Supervisor function.
Option Digit 2	This holds the 3rd Native Mode Option byte entered in the MSG MODE Configure Supervisor function.
Option Digit 3	This holds the 4th Native Mode Option byte entered in the MSG MODE Configure Supervisor function.
Option Digit 4	This holds the 5th Native Mode Option byte entered in the MSG MODE Configure Supervisor function.
Option Digit 5	This holds the 6th Native Mode Option byte entered in the MSG MODE Configure Supervisor function.
Option Digit 6	This holds the 7th Native Mode Option byte entered in the MSG MODE Configure Supervisor function.
Option Digit 7	This holds the 8th Native Mode Option byte entered in the MSG MODE Configure Supervisor function.
Option Digit 8	This holds the 9th Native Mode Option byte entered in the MSG MODE Configure Supervisor function.
Option Digit 9	This holds the 10th Native Mode Option byte entered in the MSG MODE Configure Supervisor function.
Option Digit 10	This holds the 11th Native Mode Option byte entered in the MSG MODE Configure Supervisor function.
Option Digit 11	This holds the 12th Native Mode Option byte entered in the MSG MODE Configure Supervisor function.
Option Digit 12	This holds the 13th Native Mode Option byte entered in the MSG MODE Configure Supervisor function.
Option Digit 13	This holds the 14th Native Mode Option byte entered in the MSG MODE Configure Supervisor function.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

CDI Store Worker	Description
Option Digit 14	This holds the 15th Native Mode Option byte entered in the MSG MODE Configure Supervisor function.
Option Digit 15	This holds the 16th Native Mode Option byte entered in the MSG MODE Configure Supervisor function.
Free JDATA Enable Flag	This Flag, which is set via a Terminal Command, controls access to tampered journal records using the Free Jdata function in the Supervisor Access menu. The tampered records feature is not supported.
Audio Echo Option	Non Supported Feature.
DAS Error Reporting Option	Non Supported Feature.
Option Digits String	This holds the 16 Native Mode Option Values as a string for use in the Terminal State response to the 'Send Local Configuration Option Digits' Terminal Command.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

FIT Data

Most of the CDI workers in this catalog store information from the currently matched FIT definition.

Table C-17
FIT Data

CDI Store Worker	Description
Bank ID Index	This is the offset on a card, which is combined with the PINDX to determine an Algorithm number. It is used for Diebold Local PIN Verification, which is not supported.
PAN Data Index	This is the offset on a card which is combined with the PINDX to determine the PAN.
PAN Data Length	This combines the number of digits from the PAN and a local check for a short PIN. CDI is provided for NDC+ migration.
Local Check Flag	This allows for a local check for a short PIN when Remote PIN Verification is used.
PIN Check Length	This holds the number of digits to be used from the offset PANDX for PIN Verification.
PAN Pad Option	This combines the keys required for PIN verification and PAN Pad character. CDI is provided for NDC+ migration.
PAN Pad char	This holds the PAN Pad character for padding the PAN, to the right, up to 16 digits.
PAN Pad key	This is the PIN verification Key (Master Key or PEKEY in FIT).
PIN Digits Option	This combines the PIN verification method and the number of digits to use from the PIN. CDI is provided for NDC+ migration.
PIN Digits Check Number	This is the number of digits from the PIN to check in the local PIN verification. It must be less than the maximum digits for the entered PIN.
PIN Verification Method	This determines whether the PIN verification is Remote or Local. If it is Local, the type of verification must be selected, (VISA, DES or Diebold).
Institution ID Table Entry 0	This holds the first element of Financial Institution ID as matched on the card during the FIT Search.
Institution ID Table Entry 1	This holds the second element of Financial Institution ID as matched on the card during the FIT Search.
Institution ID Table Entry 2	This holds the third element of Financial Institution ID as matched on the card during the FIT Search.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

CDI Store Worker	Description
Institution ID Table Entry 3	This holds the fourth element of Financial Institution. ID as matched on the card during the FIT Search.
Institution ID Table Entry 4	This holds the fifth element of Financial Institution ID as matched on the card during the FIT Search.
Institution ID Table Entry 5	This is not used. Only elements 0 to 4 are actually used as FIID is compressed into 5 CDIs.
Institution ID Table Entry 6	This is not used. Only elements 0 to 4 are actually used as FIID is compressed into 5 CDIs.
Institution ID Table Entry 7	This is not used. Only elements 0 to 4 are actually used as FIID is compressed into 5 CDIs.
Institution ID Table Entry 8	This is not used. Only elements 0 to 4 are actually used as FIID is compressed into 5 CDIs.
Institution ID Table Entry 9	This is not used. Only elements 0 to 4 are actually used as FIID is compressed into 5 CDIs.
Institution ID Index Option	This holds the offset on a card, which is combined with the PINDX for the Financial Institution ID, and used in the FIT Search.
Institution ID Index Location	This holds the offset on a card, combined with the PINDX, for Financial Institution ID. It is used in the FIT Search.
Institution ID Index Location Flag	This holds the offset on a card, combined with the PINDX, for Financial Institution ID. It is used in the FIT Search.
Index Reference Point Entry 0	This holds the location of the following fields (track, delimiter, search direction), PIDDX, PAGDX, PANDX, PRCNT, POFDX, PLNDX.
Index Reference Point Entry 1	This holds the location of the following fields (track, delimiter, search direction), PIDDX, PAGDX, PANDX, PRCNT, POFDX, PLNDX.
Index Reference Point Entry 2	This holds the location of the following fields (track, delimiter, search direction), PIDDX, PAGDX, PANDX, PRCNT, POFDX, PLNDX.
Index Reference Point Entry 3	This is not used. Only entries 0 to 2 are required and used, since data downloaded in FIT Table is compressed.
Index Reference Point Entry 4	This is not used. Only entries 0 to 2 are required and used, since data downloaded in FIT Table is compressed.
Index Reference Point Entry 5	This is not used. Only entries 0 to 2 are required and used, since data downloaded in FIT Table is compressed.
PIN Pad Option	This combines Pad Char and Encryption method. CDI is provided for NDC+ migration.
PIN Pad Encrypt Type	This holds the encryption method and keys to use for the creation of the PIN block.
PIN Pad Character	This holds the PIN Pad character for padding the PIN, to the right, up to 16 digits.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

CDI Store Worker	Description
Language Code Index	This holds the offset on a card, which combined with the PINDX makes up the Language Code.
PIN Block Option	This combines the data for maximum PIN digits, and PIN Block type. CDI is provided for NDC+ migration.
PIN Block Type	This holds the PIN Block type for inclusion in the Transaction Request.
PIN Block Max Digits	This holds the maximum number of digits for the PIN Entry.
PIN Offset Index	This is the offset on a card and combines with PINDX, for the PIN Offset data; or with PVKI and PVV in the VISA Pin Verification.
PIN Retry Count Option	This combines the PIN Retry Count location and whether the retry is incremented or decremented. CDI is provided for NDC+ migration.
PIN Retry Count Location	This is the offset on a card and combines with the PINDX for the PIN Retry Count.
PIN Retry Count Type	This is the Specifies Retry count mechanism, which increments or decrements the count.
GBP Reference Point	This is part of the FIT Data and is not used in Advance NDC.
Indirect Next State Index Option	This combines the data for Branch & Logo. CDI is provided for NDC+ migration.
Indirect Next State Index Location	This is used by the FIT Switch and the Expanded FIT Switch, to allow branching based the FIT definition the card is matched to.
Indirect Next State Index Logo Number	This displays the logo when the FIT dependent logo control is in a screen definition.
GBP B Index	This is part of the FIT Data and not used in Advance NDC.
GBP B Length	This is part of the FIT Data and not used in Advance NDC.
Language Code	The is the Language Code read from the card, the location of which is determined by FIT Data.
Financial Institute ID	This is not used. It is a duplicate of PFIID, the Financial Institute ID, as matched on the card during the FIT Search.
Bank ID	This is the bank ID read from a card, the location of which is determined by the FIT Data.
PIN Retry Count	This is the PIN Entry retry count read from the card, the location of which is determined by the FIT Data.
PAN	This is the Primary Account Number (PAN) read from the card, the location of which is determined by the FIT Data.
PIN Offset Data	This holds the PIN Offset data used in local PIN Verification. It is read from the card based on the specification of the FIT table.
Encrypted PE Key	The Encrypted PE Key for use in PIN verification (optional).

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

CDI Store Worker	Description
Decimalisation Table	The Decimalisation Table for the current FIT, used to convert hexadecimal data during a DES PIN verification.
Base FIT Present	The number of the FIT on which a FIT match has occurred.
FIT Match Index	A flag set TRUE when the Base FIT has been loaded, and FALSE when a Linked FIT has been loaded.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Key Entry Mode

Table C-18
Key Entry Mode

CDI Store Worker	Description
Key Entry Mode	This holds the current key entry mode configured for the terminal: 1 = Single Key Entry without XOR 2 = Single Key Entry with XOR 3 = Double Key Entry with XOR 4 = Double Key Entry—restricted The default is 1.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Misc Counters

Table C-19
Misc Counters

CDI Store Worker	Description
Camera Film Frames Remaining	This counts the number of film frames remaining. The data is used in the Send Counters Terminal Command, Supervisor Replenish functions and locations where pictures are taken. This counter is not used in Advance NDC as the security camera is not supported.
Card Captured Count	This counts the number of cards captured. The data is used in the Send Counters Terminal Command, Supervisor Replenish functions and locations where cards are captured.
Envelopes Deposited Count	This counts the number of envelopes deposited. It is used in the Send Counters Terminal Command, Supervisor Replenish functions and the Dispense Function Execution.
Statement Captured Count	This counts the number of statements captured. It is used in the Send Counters Terminal Command, Supervisor Replenish functions and the Statement Capture.
Total Transaction Count Low	See TSN and TSNBackup on page C-35. Low holds up to 10000.
Total Transaction Count High	See TSN and TSNBackup on page C-35. High holds up to 1000.
All '... Last Cleared Date' workers, for example Card Count Last Cleared Date	These are not used in Advance NDC.
Last Envelope Serial Number	This is not used in Advance NDC.
DPM Pocket/Bin Counts	These are not used; the DPM is not supported in Advance NDC.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and/or distribution is strictly prohibited.

Note Counters

Table C-20
Note Counters

CDI Store Worker	Description
Notes To Dispense Cassette *x*	This holds Transaction Reply field g, indicating the number of notes to be dispensed from cassette type*x*. If option 76 is set to 000, there will be instances for cassette types 1 to 4 If option 76 is set to 001, there are instances for cassette types 1 to
Notes Dispensed Cassette *x* Backup Count	This is a backup store of notes dispensed from cassette type *x*. If option 76 is set to 000, there will be instances for cassette types 1 to 4 If option 76 is set to 001, there are instances for cassette types 1 to
Notes Remaining Cassette *x* Backup Count	This is a backup store of notes remaining in cassette type *x*. If option 76 is set to 000, there will be instances for cassette types 1 to 4 If option 76 is set to 001, there are instances for cassette types 1 to
Notes Purged Cassette *x* Backup Count	This is a backup store of notes purged from cassette type x If option 76 is set to 000, there will be instances for cassette types 1 to 4 If option 76 is set to 001, there are instances for cassette types 1 to 7
Notes Last Dispensed Cassette *x* Count	This holds the number of notes dispensed in the last transaction from cassette type *x*. If option 76 is set to 000, there will be instances for cassette types 1 to 4 If option 76 is set to 001, there are instances for cassette types 1 to 7 It allows Central to recover the note counts in the event of a power failure. The data is used in field g7 of the Counters Message.
Notes Dispensed Cassette *x* Count	This holds the total number of notes dispensed from cassette type *x*. If option 76 is set to 000, there will be instances for cassette types 1 to 4 If option 76 is set to 001, there are instances for cassette types 1 to
Notes Remaining Cassette *x* Count	This holds the number of notes remaining in cassette type *x*. If option 76 is set to 000, there will be instances for cassette types 1 to 4 If option 76 is set to 001, there are instances for cassette types 1 to
Notes Purged Cassette *x* Count	This holds the number of notes purged from cassette type *x*. If option 76 is set to 000, there will be instances for cassette types 1 to 4 If option 76 is set to 001, there are instances for cassette types 1 to 7
Standard Note Count Cassette *x*	This holds the standard note count for cassette type *x*, set via the 'Set Cash' Supervisor Configure Menu function. If option 76 is set to 000, there will be instances for cassette types 1 to 4 If option 76 is set to 001, there are instances for cassette types 1 to
Initial Total Notes Count CDM 1	This holds the total number of notes entered for the cassette types in cash handler 1 of a dual cash handler

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

CDI Store Worker	Description
Initial Total Notes Count CDM 2	This holds the total number of notes entered for the cassette types in cash handler 2 of a dual cash handler
Cash Count Last Cleared Date	This is not used in Advance NDC.
Notes To Dispense Count	This holds the number of notes to dispense from each cassette type.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Option Digit Stores

Table C-21
Option Digit Stores

CDI Store Worker	Description
Option 48	This holds the value of option 48 For details of this option, refer to the *APTRA Advance NDC, Reference Manual*.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Printing

Table C-22
Printing

CDI Store Worker	Description
Envelope First use After Supervisor mode	This is set to TRUE if using the Envelope Dispenser for the first time after a Supervisor exit. It determines whether Unsolicited device fault messages need to be sent.
Journal Printed Flag	This is set to TRUE if printing has occurred and it is used to tidy up in the Close State.
Journal First use After Supervisor mode	This is set to TRUE if using the Journal Printer for the first time after a Supervisor exit. It determines if Unsolicited device fault messages need to be sent.
Receipt First use After Supervisor mode	This is set to TRUE if using the Receipt Printer for the first time after a Supervisor exit. It determines if Unsolicited device fault messages need to be sent.
Statement First use After Supervisor mode	This is set to TRUE if using the Statement Printer for the first time after a Supervisor exit. It determines if Unsolicited device fault messages need to be sent.
Printer Data Block 1	This holds the Transaction Reply field ‘r’ Print Data with embedded controls. The Transaction Handler processes this data and then sends it to the relevant device.
Printer Data Block 2	This holds the Transaction Reply field ‘t’ Print Data with embedded controls. The Transaction Handler processes this data and then sends it to the relevant device.
Printer Data Block 3	This holds the Transaction Reply field ‘v’ Print Data with embedded controls. The Transaction Handler processes this data and then sends it to the relevant device.
Printer Data Block 4	This holds the Transaction Reply field ‘v2’ Print Data with embedded controls. The Transaction Handler processes this data and then sends it to the relevant device.
Printer Data Block 5	This holds the Transaction Reply field ‘v4’ Print Data with embedded controls. The Transaction Handler processes this data and then sends it to the relevant device.
Printer Data Block 6	This holds the Transaction Reply field ‘v6’ Print Data with embedded controls. The Transaction Handler processes this data and then sends it to the relevant device.
Printer Data Block 7	This holds the Transaction Reply field ‘v8’ Print Data with embedded controls. The Transaction Handler processes this data and then sends it to the relevant device.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

CDI Store Worker	Description
Printer Data Block 8	This holds the Transaction Reply field ‘v10’ Print Data with embedded controls. The Transaction Handler processes this data and then sends it to the relevant device.
Printer Data Block 9	This holds the Transaction Reply field ‘v12’ Print Data with embedded controls. The Transaction Handler processes this data and then sends it to the relevant device.
Printer Data Block 10	This holds the Transaction Reply field ‘v14’ Print Data with embedded controls. The Transaction Handler processes this data and then sends it to the relevant device.
Printer Data Block 11	This holds the Transaction Reply field ‘v16’ Print Data with embedded controls. The Transaction Handler processes this data and then sends it to the relevant device.
Printer Data Block 12	This holds the Transaction Reply field ‘v18’ Print Data with embedded controls. The Transaction Handler processes this data and then sends it to the relevant device.
Printer Data Block 13	This holds the Transaction Reply field ‘v20’ Print Data with embedded controls. The Transaction Handler processes this data and then sends it to the relevant device.
Transaction Reply Printer Flags	This holds the Transaction Reply fields such as ‘s’, ‘u’, ‘v1’. It holds up to 13 printer flags specifying the devices on which each data field is to be printed. The Transaction Reply validation and Function Execution must handle many occurrences of the same Printer in the flags, matching with Func Id, and so on.
Depository Data	This holds data sent by Central in a Transaction Reply. This buffer contains the data that will be printed on the Depository.
DPM First Use After Supervisor Mode	This is not used; the DPM is not supported in Advance NDC.
Sideways Receipt Cancelled	This is set to TRUE if the sideways receipt print is cancelled.
Sideways Receipt Printed	This is set to TRUE if the sideways receipt is successfully printed.
Statement Cancelled	This is set to TRUE if the statement print is cancelled.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Screen Display

Table C-23
Screen Display

CDI Store Worker	Description
Track 1 Account Number	This is set by Track One Decomposer worker, and used when displaying the Track1 information.
Track 1 Country Code	This is set by Track One Decomposer worker, and used when displaying the Track1 information.
Track 1 Forename	This is set by Track One Decomposer worker, and used when displaying the Track1 information.
Track 1 Format	This is set by Track One Decomposer worker, and used when displaying the Track1 information.
Track 1 Optional Data	This is set by Track One Decomposer worker, and used when displaying the Track1 information.
Track 1 Surname	This is set by Track One Decomposer worker, and used when displaying the Track1 information.
Track 1 Title	This is set by Track One Decomposer worker, and used when displaying the Track1 information.
Screen Offset	This holds the offset to the first screen in the active language group.
Screen Group Size	This holds the screen group size for multi-language screens, which can be set in an 8 FDK Select or Language Select From Card State.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Security

Table C-24
Security

CDI Store Worker	Description
Time Variant Number	This holds data in the Transaction Request field ‘e’. The MAC Security Flags determine whether a Transaction Reply validation checks Transaction Reply field ‘e’ against data sent in Transaction Request.
Transaction Serial Number	This holds data from the Transaction Reply field ‘k’. It is used when taking pictures, printing on the Envelopes during a Deposit and journal traces.
Comms Key Loaded Flag	Set to TRUE if the Comms Key has been loaded. If FALSE, the locally entered Comms Key is used.
Message Coordination Number	This holds data in the Transaction Request field ‘g’. Incremented on each Transaction Request (rolls from 31H to 3FH) before inclusion and storing. In Transaction Reply validation, if the field ‘o’ in Transaction Reply is ‘0’ do not check. If non-zero, check the two values.
Last Message Status	Holds the last status issued as part of the Transaction Request field ‘r’. Set to 0 between Transaction Reply validation and Function Execution (no status sent). Set to 1 in Ready 9/B Builder (good). Set to 3 in Reject Message Builder (Transaction Reply rejected). Set to 2 in solicited device fault messages.
MAC Key Loaded Flag	Set to TRUE if the MAC Key has been loaded. If FALSE, the locally entered Comms Key is used.
NDC Security Trace Number	This holds the three digit Security Trace Number used when printing security trace messages on the Journal.
MAC Machine Number	This holds the six digit Security Terminal Number entered in the Enter MAC Access Supervisor function.
MAC Security Flags	This holds the ten Security Flags entered in the Enter MAC Access Supervisor function.
Other Message MAC Field Selection	This holds the MAC field selection for FIT Load, State Table Load, Terminal State and Dispenser Currency Cassette Mapping Table messages.
Solicited Status MAC Field Selection	This holds the MAC field selection for Solicited Status Messages.
Track 1 MAC Field Selection	This holds the MAC field selection for Track 1 data in Transaction Request and Transaction Reply.
Track 2 MAC Field Selection	This holds the MAC field selection for Track 2 data in Transaction Request and Transaction Reply.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and/or distribution is strictly prohibited.

CDI Store Worker	Description
Track 3 MAC Field Selection	This holds the MAC field selection for Track 3 data in Transaction Request and Transaction Reply.
Transaction Reply MAC Field Selection	This holds the MAC field selection for Transaction Reply fields.
Transaction Request MAC Field Selection	This holds the MAC field selection for Transaction Request fields.
TSN	This holds the last Transaction Serial Number, which is sent as part of the response to the Send Counters Terminal Command, and is part of field ‘r’ in Transaction Request.
TSN Backup	This is the backup store of idTSN, catering for power failures during the Transaction Reply Function execution.
VISA Key A0	This holds the VISA Key Table Data. It is downloaded using the Encryption Key Change Command, and is used in the PIN Entry States when local VISA verification is specified.
VISA Key A1	This holds the VISA Key Table Data. It is downloaded using the Encryption Key Change Command, and is used in the PIN Entry States when local VISA verification is specified.
VISA Key A2	This holds the VISA Key Table Data. It is downloaded using the Encryption Key Change Command, and is used in PIN Entry States when the local VISA verification is specified.
VISA Key A3	This holds the VISA Key Table Data. It is downloaded using the Encryption Key Change Command, and is used in PIN Entry States when the local VISA verification is specified.
VISA Key A4	This holds the VISA Key Table Data. It is downloaded using the Encryption Key Change Command, and is used in PIN Entry States when the local VISA verification is specified.
VISA Key A5	This holds the VISA Key Table Data. It is downloaded using the Encryption Key Change Command, and is used in PIN Entry States when the local VISA verification is specified.
VISA Key B0	This holds the VISA Key Table Data. It is downloaded using the Encryption Key Change Command, and is used in PIN Entry States when the local VISA verification is specified.
VISA Key B1	This holds the VISA Key Table Data. It is downloaded using the Encryption Key Change Command, and is used in PIN Entry States when the local VISA verification is specified.
VISA Key B2	This holds the VISA Key Table Data. It is downloaded using the Encryption Key Change Command, and is used in PIN Entry States when the local VISA verification is specified.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

CDI Store Worker	Description
VISA Key B3	This holds the VISA Key Table Data. It is downloaded using the Encryption Key Change Command, and is used in PIN Entry States when the local VISA verification is specified.
VISA Key B4	This holds the VISA Key Table Data. It is downloaded using the Encryption Key Change Command, and is used in PIN Entry States when the local VISA verification is specified.
VISA Key B5	This holds the VISA Key Table Data. It is downloaded using the Encryption Key Change Command, and is used in PIN Entry States when the local VISA verification is specified.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

State Information

Table C-25
State Information

CDI Store Worker	Description
Current State Number	Number of the State about to be executed.
Current State Type	State Type Id of the State about to be executed.
Primary Extension State Entry 0	Holds first parameter for first extension State.
Primary Extension State Entry 1	Holds second parameter for first extension State.
Primary Extension State Entry 2	Holds third parameter for first extension State.
Primary Extension State Entry 3	Holds fourth parameter for first extension State.
Primary Extension State Entry 4	Holds fifth parameter for first extension State.
Primary Extension State Entry 5	Holds sixth parameter for first extension State.
Primary Extension State Entry 6	Holds seventh parameter for first extension State.
Primary Extension State Entry 7	Holds eight parameter for first extension State.
Secondary Extension State Entry 0	Holds first parameter for second extension State.
Secondary Extension State Entry 1	Holds second parameter for second extension State.
Secondary Extension State Entry 2	Holds third parameter for second extension State.
Secondary Extension State Entry 3	Holds fourth parameter for second extension State.
Secondary Extension State Entry 4	Holds fifth parameter for second extension State.
Secondary Extension State Entry 5	Holds sixth parameter for second extension State.
Secondary Extension State Entry 6	Holds seventh parameter for second extension State.
Secondary Extension State Entry 7	Holds eight parameter for second extension State.
State Table Entry 0	Holds first parameter of main State Table entry (Table entry 2).
State Table Entry 1	Holds second parameter of main State Table entry (Table entry 3).
State Table Entry 2	Holds third parameter of main State Table entry (Table entry 4).
State Table Entry 3	Holds fourth parameter of main State Table entry (table entry 5).
State Table Entry 4	Holds fifth parameter of main State Table entry (Table entry 6).
State Table Entry 5	Holds sixth parameter of main State Table entry (Table entry 7).
State Table Entry 6	Holds seventh parameter of main State Table entry (Table entry 8).
State Table Entry 7	Holds eight parameter of main State Table entry (Table entry 9).

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

CDI Store Worker	Description
Tertiary Extension State Entry 0	Holds first parameter for third extension State.
Tertiary Extension State Entry 1	Holds second parameter for third extension State.
Tertiary Extension State Entry 2	Holds third parameter for third extension State.
Tertiary Extension State Entry 3	Holds fourth parameter for third extension State.
Tertiary Extension State Entry 4	Holds fifth parameter for third extension State.
Tertiary Extension State Entry 5	Holds sixth parameter for third extension State.
Tertiary Extension State Entry 6	Holds seventh parameter for third extension State.
Tertiary Extension State Entry 7	Holds eight parameter for third extension State.
Fourth Extension State Entry 0	Holds first parameter for fourth extension State.
Fourth Extension State Entry 1	Holds second parameter for fourth extension State.
Fourth Extension State Entry 2	Holds third parameter for fourth extension State.
Fourth Extension State Entry 3	Holds fourth parameter for fourth extension State.
Fourth Extension State Entry 4	Holds fifth parameter for fourth extension State.
Fourth Extension State Entry 5	Holds sixth parameter for fourth extension State.
Fourth Extension State Entry 6	Holds seventh parameter for fourth extension State.
Fourth Extension State Entry 7	Holds eight parameter for fourth extension State.
Fifth Extension State Entry 0	Holds first parameter for fifth extension State.
Fifth Extension State Entry 1	Holds second parameter for fifth extension State.
Fifth Extension State Entry 2	Holds third parameter for fifth extension State.
Fifth Extension State Entry 3	Holds fourth parameter for fifth extension State.
Fifth Extension State Entry 4	Holds fifth parameter for fifth extension State.
Fifth Extension State Entry 5	Holds sixth parameter for fifth extension State.
Fifth Extension State Entry 6	Holds seventh parameter for fifth extension State.
Fifth Extension State Entry 7	Holds eight parameter for fifth extension State.
Sixth Extension State Entry 0	Holds first parameter for sixth extension State.
Sixth Extension State Entry 1	Holds second parameter for sixth extension State.
Sixth Extension State Entry 2	Holds third parameter for sixth extension State.
Sixth Extension State Entry 3	Holds fourth parameter for sixth extension State.
Sixth Extension State Entry 4	Holds fifth parameter for sixth extension State.
Sixth Extension State Entry 5	Holds sixth parameter for sixth extension State.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

CDI Store Worker	Description
Sixth Extension State Entry 6	Holds seventh parameter for sixth extension State.
Sixth Extension State Entry 7	Holds eight parameter for sixth extension State.
Seventh Extension State Entry 0	Holds first parameter for seventh extension State.
Seventh Extension State Entry 1	Holds second parameter for seventh extension State.
Seventh Extension State Entry 2	Holds third parameter for seventh extension State.
Seventh Extension State Entry 3	Holds fourth parameter for seventh extension State.
Seventh Extension State Entry 4	Holds fifth parameter for seventh extension State.
Seventh Extension State Entry 5	Holds sixth parameter for seventh extension State.
Seventh Extension State Entry 6	Holds seventh parameter for seventh extension State.
Seventh Extension State Entry 7	Holds eight parameter for seventh extension State.
Eighth Extension State Entry 0	Holds first parameter for eighth extension State.
Eighth Extension State Entry 1	Holds second parameter for eighth extension State.
Eighth Extension State Entry 2	Holds third parameter for eighth extension State.
Eighth Extension State Entry 3	Holds fourth parameter for eighth extension State.
Eighth Extension State Entry 4	Holds fifth parameter for eighth extension State.
Eighth Extension State Entry 5	Holds sixth parameter for eighth extension State.
Eighth Extension State Entry 6	Holds seventh parameter for eighth extension State.
Eighth Extension State Entry 7	Holds eight parameter for eighth extension State.
Current Mode	Determines the Current Mode of the SST.
PIN Entry Status	Used to determine the outcome of PIN Entry initiation in a Card Read State. Values are 0 = PIN Good, 1 = PIN Cancelled, 2 = PIN Timeout.
PIN Entry Initiated	Used to determine if the previous state was a PIN Entry Initiation state. Only used by PIN Entry State.
Current Mode (NVRAM)	Used to determine the current Supervisor mode. INITIALISE = 0, POWERUP = 1, OOS = 2, SUPPLY / SUPERVISOR = 3, IN SERVICE = 4, OFFLINE =5, SUSPEND = 6.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Supervisor

Table C-26
Supervisor

CDI Store Worker	Description
Envelope Dispenser Status	This holds the Status of the Envelope Dispenser, and is used to determine if an Unsolicited Device Status message needs to be sent to Central.
Card Capture Bin Status	This holds the Status of the Card Capture Bin, and is used to determine if an Unsolicited Device Status message needs to be sent to Central.
Receipt Paper Status	This holds the Status of the Receipt Paper, and is used to determine if an Unsolicited Device Status message needs to be sent to Central.
Receipt Ribbon Status	This holds the Status of the Receipt Ribbon, and is used to determine if an Unsolicited Device Status message needs to be sent to Central.
Receipt Printhead Status	This holds the Status of the Receipt Printhead, and is used to determine if an Unsolicited Device Status message needs to be sent to Central.
Receipt Knife Status	This holds the Status of the Receipt Knife, and is used to determine if an Unsolicited Device Status message needs to be sent to Central.
Journal Paper Status	This holds the Status of the Journal Paper, and is used to determine if an Unsolicited Device Status message needs to be sent to Central.
Journal Ribbon Status	This holds the Status of the Journal Ribbon, and is used to determine if an Unsolicited Device Status message needs to be sent to Central.
Journal Printhead Status	This holds the Status of the Journal Printhead, and is used to determine if an Unsolicited Device Status message needs to be sent to Central.
Statement Paper Status	This holds the Status of the Statement Paper, and is used to determine if an Unsolicited Device Status message needs to be sent to Central.
Statement Ribbon Status	This holds the Status of the Statement Ribbon, and is used to determine if an Unsolicited Device Status message needs to be sent to Central.
Statement Printhead Status	This holds the Status of the Statement Printhead, and is used to determine if an Unsolicited Device Status message needs to be sent to Central.
Statement Knife Status	This holds the Status of the Statement Knife, and is used to determine if an Unsolicited Device Status message needs to be sent to Central.
Statement Capture Bin Status	This holds the Status of the Statement Capture Bin, and is used to determine if an Unsolicited Device Status message needs to be sent to Central.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

CDI Store Worker	Description
Return Mode	This determines which mode to return to when exiting from Supervisor - previous mode or Out of Service. It is configured using a downloaded Configuration Parameter or Enhanced Configuration Parameter (Option 1).
Supervisor Password	This holds the 4-digit (numeric) password which, if configured, controls entry to the Supervisor Configure and Access menus. The password is configured using a function in the Access Menu.
Initial Supervisor Menu	This specifies which of the following menus is to be displayed when entering the Supervisor mode: ● 0 - Select ● 1 - Replenish ● 2 - Exit Supervisor. This is configured using a function in the Supervisor Access menu.
Initial Supervisor Display	This specifies which of the following interfaces is to be used on entry to Supervisor mode: ● 0 - default ● 1 - front. It is configured using a function in the Supervisor Access menu.
Supervisor Password Flag	This Flag indicates whether or not a Supervisor password has been configured, allowing for a controlled entry to the Configure and Access menus.
Exit Mode	If this is 0, the Supervisor mode switch plus entering 9 on the Select Supervisor menu is needed to exit Supervisor. If it is 1, only the mode switch is needed to exit Supervisor. This is defined in the Exit Mode Supervisor Access Menu function.
Default Printer	This holds the Default Printer to use for the PRNT CMPNT VERS, PRNT SCRW VERS, PRNT SPPLY, PRNT CNTRS, PRNT CONFIG, PRNT TCP/IP, and PRNT ACC Supervisor functions. This is defined in the Set Print Supervisor Configure Menu function.
Restart Mode	This determines the recovery after power failure/reset. The value is entered in the Restart Mode Supervisor Configure Menu function.
Return Mode (NVRAM)	This is used at power-up to determine the mode to return to after exiting from Supervisor mode.
Supervisor Password Retry	This Counter keeps track of the number of attempts there has been to enter the Supervisor password which controls the entry to the Configure and Access menus. Five attempts are allowed at the front interface, but only one attempt at the rear interface.
EJ Log Sequence Number	This is used in the filename extension of the backup EJ log file.
Depository Bin Status	This holds the Status of the Depository Bin, and is used to determine if an Unsolicited Device Status message needs to be sent to Central.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and/or distribution is strictly prohibited.

CDI Store Worker	Description
Journal Knife Status	This holds the Status of the Journal Knife, which is always ‘good’ as there is no journal knife.
Set Access	This is used to write the Supervisor access configuration to disk.
Set Default	This is used to write the Supervisor defaults to disk.
DPM Enable Image Dump Flag	This is not used; the DPM is not supported in Advance NDC.
Supervisor Entry Message (pending)	This is used to contain the unsolicited Supervisor Keys message indicating entry to Supervisor mode.
Supervisor Exit Message (pending)	This is used to contain the unsolicited Supervisor Keys message indicating exit from Supervisor mode.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Terminal Configuration

Table C-27
Terminal Configuration

CDI Store Worker	Description
PAN DCS Option	This specifies whether or not to include PAN in the data sent to the DCS.
Product ID	This holds the Product Id which is reported back within the Terminal State response to a Send Configuration Information Terminal Command.
Auto Voice Option	This is an Enhanced Configuration Parameter (Option 2). If set to TRUE, voice message 1 is played at the same time as the please wait screen. It is used to control the playing of fixed message numbers.
Beep at Card Eject	This is an Option Digit (part of 4). If set to 0, a beep will sound when the card is ejected.
Beep at Cash Present	This is an Option Digit (part of 2). If set to 1, a beep will sound when presenting cash during the Cash Dispense Transaction Reply Function Execution.
Beep at Deposit	This is an Option Digit (part of 2). If set to 0, a beep will sound when depositing an envelope during the Deposit Transaction Reply Function Execution.
Camera NAK Reporting Option	This is an Option Digit (part of 3). It determines if NAK from the camera is reported to Central.
Camera Control Option	This is a Configuration Parameter or an Enhanced Configuration Parameter (Option 0). If a valid Transaction Reply is received and this flag is set to 0 or 1, a picture is taken automatically.
Cancel/Clear Swap Option	This is an Option Digit (part of 7). It determines if the cancel/clear keys are swapped. (Not supported in Advance NDC)
Cancel On Statement and Wait Option	This is an Option Digit (part of 3). If set to 1, the Cancel key is enabled during the Statement Print and Wait Transaction Reply.
Card Captured Trace Option	This is an Option Digit (part of 5). If set to 1, data is printed on the Journal when a card is captured.
Card Read Error Threshold	This is a Configuration Parameter or an Enhanced Configuration Parameter (Option 13). It is not used as it is a Diebold-only feature.
Night Safe Depository Check Flag	This is an Option Digit (part of 3). It determines when to check the Night Safe for an overfill condition.
Comms Status Trace Option	This is an Option Digit (part of 5). If set to 1, tracing is done on the Journal when the Communications goes On/Off line.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and/or distribution is strictly prohibited.

CDI Store Worker	Description
Date Format Option	This is an Enhanced Configuration Parameter (Option 3) which specifies the format of the date when using it for tasks such as printing.
Dispense Status Delay Option	This is an Option Digit (part of 4), which controls when the Dispenser solicited status messages are sent.
Download Save Option	This is an Option Digit (part of 5) and determines when any downloaded data is saved to disk.
Hard Copy Backup Time Limit	This is an Enhanced Configuration Parameter (option 16) holding the maximum time for the Journal backup when Printer is non-operational.
Hard Copy Backup Records Limit	This is an Enhanced Configuration Parameter (option 17) holding the maximum number of records to backup for the Journal backup when Printer is non-operational.
Envelope Dispenser Status Reporting Option	This is an Enhanced Configuration Parameter (Option 23) and determines whether the Envelope Dispenser status messages should be sent to Central.
Extended Status Reporting Option	This is an Enhanced Configuration Parameter (Option 11). It is not used as it is a Diebold only feature.
Journal Mode	This determines where the Journal output is to be directed.
Left / Right FDK Swap Option	This is an Option Digit (part of 7) and determines whether the Left/Right FDKs are to be used in Keyboard Entry States.
Left Print Column	This holds the value for the left margin for the Receipt and the Journal. It is entered in either the Roll Width Configure Supervisor function, or set in an Enhanced Configuration Parameters download, (Option 5). The conversion process will not modify this flag.
Max Receipt Lines	This is an Option Digit (part of 6) and specifies the maximum number of receipt lines; 49 if TRUE, or 24. The standard receipt length for Advance NDC is 24 lines. Applications must add a FF and CUT after 24 lines (or 49 if `idMaxReceiptLines` is set). The conversion process will not modify this flag.
MEI Flash Rate Option	This is an Enhanced Configuration Parameter (Option 25), or Option Digit (8), and controls the flash rate of the Media Entry Indicators.
Product Class	This stores the product class, as reported in the Hardware Configuration Terminal Command.
Ready B Option	This is a Configuration Parameter or Enhanced Configuration Parameter (part of Option 1). If it is set to TRUE, after a successful Transaction Reply Function Execution a Ready B message is sent instead of a Ready 9.
Remote Relay Option	This is an Enhanced Configuration Parameter (Option 27), and determines when the Remote relay is active.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and/or distribution is strictly prohibited.

CDI Store Worker	Description
Retract Untaken Notes Option	This is an Option Digit (Part of 2), and determines whether any notes not taken should be retracted during the Cash Dispense Transaction Reply Function Execution.
Software ID Option	This is an Option Digit (part of 6), and determines if Advance NDC release number and S/W ID are in the Configuration Terminal State messages.
Specific Command Reject Option	This is an Enhanced Configuration Parameter (Option 12). If execution of an incoming message fails, this flag determines if Command Reject or Specific Command Reject is to be used.
Supervisor Key Reporting Option	This is an Option Digit (0), and determines which Supervisor Key presses are reported to Central.
Track 1 Format Option	This is an Enhanced Configuration Parameter (Option 7), and holds the format for Track 1. This can also be set via the Track 1 Configure Supervisor function.
Transmit TI Status Option	This is an Enhanced Configuration Parameter (Option 10), used for transmitting TI Status. It is not used as it is a Diebold only feature.
TI Sensor Status Reporting Option	This is an Enhanced Configuration Parameter (Option 24), for determining whether enhanced TI/Sensor status unsolicited status messages are sent.
Security Trace Option	This flag is set using the Trace On/Off Configure Supervisor functions. It determines if Communications tracing on the Journal Printer is active.
Transaction Status Information Option	This is an Enhanced Configuration Parameter (Option 15), used for the inclusion of Transaction Status Information in the Transaction Request message.
Roll Width	This defines the number of columns to print on for the Receipt and Journal printers. The information is entered in the Roll Width Configure Supervisor function or set in an Enhanced Configuration Parameter (Option 4). The conversion process will not modify this flag.
Supervisor Auto Return	This is a Configuration Parameter or Enhanced Configuration Parameter (Part of Option 1), and determines whether or not to return to the previous mode on exiting from Supervisor mode.
Configuration ID	This holds the Configuration ID downloaded in a Configuration ID Load message.
Current Version	This holds the Current Release Number of Advance NDC. It is used in the Terminal State message in response to the Send Configuration Information Terminal Command.
Logical Unit Number	This holds the LUNO (Logical Unit Number) which is present in many NDC+ Communications messages.
Machine Number	This holds the Machine Number which is entered in the Machine No Configure Supervisor function. It is used when taking camera pictures, in Deposit printing and in the EJ backup log filename.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and/or distribution is strictly prohibited.

CDI Store Worker	Description
Configuration Saved	This is a flag, used to indicate that the configuration data has been saved.
MM Option	This is not supported in Advance NDC.
Touch Screen Error Reporting Option	This is not supported in Advance NDC.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Timers

Table C-28
Timers

CDI Store Worker	Description
Timer 0 Value	This is a Configuration Parameter or an Enhanced Configuration Parameter and holds the value for Timer 0.
Timer 1 Value	This is a Configuration Parameter or an Enhanced Configuration Parameter and holds the value for Timer 1.
Timer 2 Value	This is a Configuration Parameter or an Enhanced Configuration Parameter and holds the value for Timer 2.
Timer 3 Value	This is a Configuration Parameter or an Enhanced Configuration Parameter and holds the value for Timer 3.
Timer 4 Value	This is a Configuration Parameter or an Enhanced Configuration Parameter and holds the value for Timer 4.
Timer 5 Value	This is a Configuration Parameter or an Enhanced Configuration Parameter and holds the value for Timer 5.
Timer 6 Value	This is a Configuration Parameter or an Enhanced Configuration Parameter and holds the value for Timer 6.
Timer 7 Value	This is a Configuration Parameter or an Enhanced Configuration Parameter and holds the value for Timer 7.
Timer 8 Value	This is a Configuration Parameter or an Enhanced Configuration Parameter and holds the value for Timer 8.
Timer 9 Value	This is a Configuration Parameter or an Enhanced Configuration Parameter and holds the value for Timer 9.
Timer 10 Value	This is a Configuration Parameter or an Enhanced Configuration Parameter and holds the value for Timer 10.
Timer 61 Value	This is an Enhanced Configuration Parameter and holds the value for Timer 61 (barcode reader scan Timer)
Timer 68 Value	This is an Enhanced Configuration Parameter and holds the value for Timer 68 (Statement Media Entry / Exit Indicator Duration Timer).
Timer 69 Value	This is an Enhanced Configuration Parameter and holds the value for Timer 69 (Receipt Media Entry / Exit Indicator Duration Timer).
Timer 72 Value	This is an Enhanced Configuration Parameter and holds the value for Timer 72 (remove card timer for DASH cards).
Timer 77 Value	This is an Enhanced Configuration Parameter and holds the value for Timer 77 (Cash Acceptance Timer for the BNA).

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

CDI Store Worker	Description
Timer 82 Value	This is a Configuration Parameter or an Enhanced Configuration Parameter and holds the value for Timer 82.
Timer 83 Value	This is a Configuration Parameter or an Enhanced Configuration Parameter and holds the value for Timer 83.
Timer 86 Value	This is a Configuration Parameter or an Enhanced Configuration Parameter and holds the value for Timer 86.
Timer 87 Value	This is a Configuration Parameter or an Enhanced Configuration Parameter and holds the value for Timer 87.
Timer 91 Value	This is a Configuration Parameter or an Enhanced Configuration Parameter and holds the value for Timer 91.
Timer 92 Value	This is a Configuration Parameter or an Enhanced Configuration Parameter and holds the value for Timer 92.
Timer 93 Value	This is a Configuration Parameter or an Enhanced Configuration Parameter and holds the value for Timer 93.
Timer 94 Value	This is a Configuration Parameter or an Enhanced Configuration Parameter and holds the value for Timer 94.
Timer 95 Value	This is a Configuration Parameter or an Enhanced Configuration Parameter and holds the value for Timer 95.
Timer 96 Value	This is a Configuration Parameter or an Enhanced Configuration Parameter and holds the value for Timer 96.
Timer 97 Value	This is a Configuration Parameter or an Enhanced Configuration Parameter and holds the value for Timer 97.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Transaction Processing Flags

Table C-29
Transaction Processing Flags

CDI Store Worker	Description
Card Jammed Flag	This is set to TRUE if a Card is jammed. It is not used in Advance NDC.
Return / Retain Card Flag	This holds the Transaction Reply field 'p', and determines either to eject or capture the card at the end of a transaction. It is used in the Close State to eject or capture cards.
Envelope Dispensed Flag	This indicates if an envelope has been presented in an Envelope Dispense State. It determines whether there is a need to dispense an envelope in the Deposit Function.
Envelope Presented Flag	This indicates if an envelope has been presented in an Envelope Dispense State. It determines whether there is a need to dispense an envelope in the Deposit Function.
FIT Match Found Flag	This is set to TRUE in the Card Read State, if a FIT Match has been found. It is used in State Types such as the FIT Switch.
Track Language Offset	This holds the offset on the Card Track where the language code can be found. It is not used as a duplicate of the PLNDX.
Next State Number	This holds the Next State Number to be executed. This is set by each State Type on completion determining the next State to be executed.
Receipt Printed Flag	If set to TRUE then printing has occurred. It is used to tidy up in the Close State.
Top of Receipt Flag	This holds the Transaction Request field 'f', and is used in the Transaction Reply Printer handling. It is set to TRUE on startup. In the Close State, if it is set to FALSE and the receipt is printed on, then the receipt is delivered / cut and the flag is set to TRUE. This flag is true if the receipt printer head is at the top of the page. The conversion process will set this flag depending on the value of `TRPrintLineCount`, Message Mode Option 6, and whether the last item processed was a Line Feed sequence. If this flag is set to FALSE in the Close State then the `mNDCPrintFooter` will cut and eject a receipt.
Statement Printed Flag	If it is set to TRUE then printing has occurred. It is used to tidy up in the Close State.
Track 1 Update Flag	This is set to FALSE in the Transaction Reply initialisation. It is set to TRUE if there is T1 data in the Transaction Reply. It is used in the Card Write State.
Track 2 Update Flag	This is set to FALSE in the Transaction Reply initialisation. It is set to TRUE if there is T2 data in the Transaction Reply. It is used in the Card Write State.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

CDI Store Worker	Description
Track 3 Update Flag	This is set to FALSE in the Transaction Reply initialisation. It is set to TRUE if there is T3 data in the Transaction Reply. It is used in the Card Write State.
Transaction Reply Function ID	This holds the Transaction Reply field 'l' - function to execute. It influences the validation of data, controls the function to execute and is used in the Status Handling.
Customisation Layer Status	This holds data for the Status of the Customisation Layer. It is used to determine what processing of messages, particularly Transaction Reply, is required in the Application Core.
Power Fail Recovery Flag	This holds data for the Power Fail Recovery Flag. It is set before the Transaction Reply Function Execution, and is checked on start up to see whether a power fail has occurred during a transaction.
Top of Statement Flag	This is used in the Transaction Reply Printer handling. It is set to TRUE on startup. In the Close State, if it is set to FALSE and the Statement is printed on, then deliver/cut statement and set flag to TRUE.
Dialup Flag	Used to indicate that the SST communicates with Central through dialup communications (VISA2)
CAV Started Flag	This is not used in Advance NDC.
Document Presented Flag	This is not used in Advance NDC.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and/or distribution is strictly prohibited.

UCDI Stores

Table C-30
UCDI Stores

UCDI Store Worker	Description
HostMessage	Contains a message received from Central.
MessageForHost	Stores a new 'user' message for sending to Central.
StatusInfo	Stores the status information used in a terminal state message.
PendingTC	Stores a Pending Terminal Command.
PendingTR	Stores a Pending Transaction Request.
CPM cheque present	The Cheque Processing Module (CPM) Cheque Accept State ('w') sets this store when a cheque has been successfully entered into the CPM. This store is checked at start of day and at the end of a transaction, to determine if a cheque is present in the CPM. The supervisor Replenish menu option 43 (INIT DEPOS) clears this store.

You must ensure all UCDI stores are cleared, initialised and set by your application as required.

Note: CDI and UCDI stores used for customising printers are used only for other vendors' SSTs. For details, refer to the *APTRA Advance NDC, Multi-Vendor Support Reference Manual.*

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Appendix D

Font Definition

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Overview

This appendix describes how you can define and use your own TrueType fonts for use with screens and printers.

The appendix includes examples of the font definition and mapping files.

Note: To enable an EMV Integrated Circuit Card (ICC), also called a smart card, to be used, additional fonts and font designators are provided with Advance NDC. If you wish to use an EMV ICC, refer to the *EMV Integrated Circuit Card (ICC) Reference Manual* provided with the EMV / CAM2 Exits for APTRA Advance NDC product. The *EMV ICC Reference Manual* lists the additional EMV fonts and font designators.

This appendix also describes how to install a font after it has been created or edited, and how to check that it is usable.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Defining Fonts

Advance NDC recognises some default font names and designators, as shown in Table D-3. These default settings can be changed, as shown in the “Font Definition File” section, or additional fonts can be added using the unassigned designators listed in the table.

The next sections provide the following information:

- Creating your own fonts for screen display and printing
- Using the Font Definition File to assign fonts to the font designators used in Advance NDC
- Adjusting your downloadable fonts for a different resolution printer
- Installing fonts
- Checking fonts.

Creating Fonts

You can extend the default fonts supported by Advance NDC using the following methods:

- Using an existing TrueType font
- Editing an existing TrueType font
- Creating your own TrueType font using a third-party utility.

For example, a utility such as ScanFont (available through the web site *www.fontlab.com*) is useful as it enables you to move glyphs within a font table, and hence change the character displayed for a particular Advance NDC character code.

You can use any fonts or font development utility you prefer.

Further details, utilities and fonts can be found on the Microsoft Typography web site *www.microsoft.com/typography*.

Font Definition File

The font definition file is a user-created file that defines the fonts used by Advance NDC. It can also be used to re-assign the default designators to different fonts. The file uses comma-delimited positional fields, as explained in Table D-2.

The font definition file is only required if adding additional fonts, or assigning a different font to one of the designators already recognised by Advance NDC. For further information, see the “Font Designators” section.

Fonts are selected in Advance NDC screens by using the escape sequences shown in Table D-1. In these escape sequences, *X* is the font designator. The font designator used must be either a default

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

font supported by Advance NDC or one that you have installed and included in the font definition file.

Table D-1
Font Selection Escape Sequences

Escape Sequence	Description
ESC (*X*	Selects the font to use for the primary character set
ESC) *X*	Selects the font to use for the secondary character set

The font definition file *fontdefs.txt* must be created and installed to *c:\program files\ncr aptra\advance ndc\data* if you want to define additional or redefine default fonts. The file must contain a line for additional or redefined font, in the following format:

```
Font Designator, Font Face, Char Set, Width, Height, X Offset, Y Offset, Double Height, Bold, Italic, Underline, Strikeout
```

These fields are defined in Table D-2. All the fields must be in order, on one line with no word wrapping. There must not be any blank lines in the file.

Table D-2
Fields in the Font Definition File

Field	**Description**
Font Designator	The symbol identifying the font. See Table D-3 for the font designators and any fonts assigned by default. If no font designator is specified, 1 (NDC Alphanumeric 1) is used.
Font Face Name	The name of the font face. See Table D-3 for the font face names with their default designations.
Char Set	The name or value of the character set. See Table D-4 for a list of accepted character set names and values.
Width	The width of the characters in logical units according to the aspect ratio of the selected font. By default, 100 for normal width fonts, 200 for double width fonts.
Height	The height of the characters in logical units according to the character cell height minus the internal-leading value. By default, 100 for normal height fonts, 200 for double height fonts.
X offset	The initial horizontal position of the text relative to the default starting position By default, 0, at the left of the display or page. A positive offset positions a character to the right. A negative offset positions a character to the left.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Field	Description
Y offset	The initial vertical position of the text relative to the default starting position By default, 0, at the top of the display or page. A positive offset positions a character up. A negative offset positions a character down.
Double Height	Whether to use double height font 0 - Do not use double height font (default) 1 - Use double height font
Bold	Whether to use bold font 0 - Do not use bold font (default) 1 - Use bold font
Italic	Whether to use italic font 0 - Do not use italic font (default) 1 - Use italic font
Underline	Whether to use underline font 0 - Do not use underline font (default) 1 - Use underline font
Strikeout	Whether to use strikeout 0 - Do not use strikeout (default) 1 - Use strikeout

Font Designators

Table D-3 shows the font designators recognised by Advance NDC by default without the need for a font definition file. If a font definition file is used, additional fonts can be assigned to both the unassigned and default designators. If a new font is assigned to a default designator using a font definition file, the new font is displayed whenever the designator is used.

Note 1: If no font designator is defined, the NDC Alphanumeric 1 font is used.

Note 2: Any defined fonts must be successfully installed. If they are not installed, the NDC Alphanumeric 1 font is used.

Table D-3
Font Designators

Designator	Default Font Face Name
1	NDC Alphanumeric 1
2	NDC Alphanumeric 2
3	NDC Standard Graphics 1
4	NDC Standard Graphics 2

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Designator	Default Font Face Name
5	NDC Standard Graphics 3
6	None (NDC Alphanumeric 1)
7	NDC Customer Graphics 2 (Arabic)
8	None (NDC Alphanumeric 1)
9	None (NDC Alphanumeric 1)
:	Double Size NDC Chinese 1
;	Double Size NDC Chinese 2
<	None (NDC Alphanumeric 1)
=	None (NDC Alphanumeric 1)
>	Double Size NDC Alphanumeric 1
?	Double Size NDC Alphanumeric 2
@	None (NDC Alphanumeric 1)
A	None (NDC Alphanumeric 1)
B	Double Size NDC Customer Graphics 2 (Arabic)
C	None (NDC Alphanumeric 1)
D	None (NDC Alphanumeric 1)
E	None (NDC Alphanumeric 1)
F	None (NDC Alphanumeric 1)

Character Sets

Table D-4 lists the character sets supported by Microsoft Windows that can be used within the font definition file. Either the character set name or value can be used in the font definition file to identify the character set to use.

Table D-4
Character Set Names and Values

Character Set Name	Value
ANSI_CHARSET	0
DEFAULT_CHARSET	1
SYMBOL_CHARSET	2
SHIFTJIS_CHARSET	128
HANGEUL_CHARSET	129
HANGUL_CHARSET	129

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Character Set Name	Value
GB2312_CHARSET	134
CHINESEBIG5_CHARSET	136
JOHAB_CHARSET	130
GREEK_CHARSET	161
TURKISH_CHARSET	162
VIETNAMESE_CHARSET	163
HEBREW_CHARSET	177
ARABIC_CHARSET	178
RUSSIAN_CHARSET	204
THAI_CHARSET	222
EASTEUROPE_CHARSET	238

Note: If an unsupported character set is requested, Advance NDC uses the UNKNOWN_CHARSET character set.

For details of the default character sets provided on displays and printers, refer to the *APTRA Advance NDC, Reference Manual*.

Example Font Definition File

An example Font Definition File is given in Figure D-1l. The following rules apply to the file:

- '//' characters at the beginning of a line indicate that the line is a comment
- There must not be any blank lines in the file
- Blank fields can be omitted from a line, but must be included if followed by other non-default fields
- There must not be any text wrapping.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and/or distribution is strictly prohibited.

Figure D-1
Example Font Definition File

```
// Example Font Definition File
// Format:
// Designator, Face, Set, Width, Height, X, Y, Dht, B, I, U, S/O
1, NDC ALPHANUMERIC 1, ANSI_CHARSET, 100, 100, 0, 6, 0, 0, 1, 0, 0
2, NDC ALPHANUMERIC 1, ANSI_CHARSET, 100, 100, 0, 6, 0, 0, 0, 0, 0
3, NDC ALPHANUMERIC 1, 0, 100, 100, 0, 6, 1
6, NDC ALPHANUMERIC 1, ANSI_CHARSET, 100, 100, 0, 6, 0, 0, 0, 0, 1
7, NDC ALPHANUMERIC 1, ANSI_CHARSET, 100, 100, 0, 6, 0, 0, 0, 1, 0
8, NDC ALPHANUMERIC 1, ANSI_CHARSET, 100, 100, 0, 6, 0, 1, 0, 0, 0
:, NDC ALPHANUMERIC 1, ANSI_CHARSET, 100, 100, 0, 6, 0, 0, 0, 0, 0
;, NDC ALPHANUMERIC 1, ANSI_CHARSET, 100, 100, 6, 0, 0, 0, 0, 0, 0
>, NDC ALPHANUMERIC 1, ANSI_CHARSET, 200, 100, 0, 0, 0, 0, 0, 0, 0
?, NDC ALPHANUMERIC 1, ANSI_CHARSET, 100, 200, 0, 0, 0, 0, 0, 0, 0
@, NDC ALPHANUMERIC 1, GREEK_CHARSET, 100, 100, 0, 6, 0, 1, 0, 0, 0
A, COURIER NEW, ANSI_CHARSET, 100, 100, 0, 6, 0, 0, 0, 1, 0
```

Table D-5 walks through the example definition file content following the comment lines. For further details of each field, see Table D-2.

Note: Although any font description can be set out in the font definition file, not all character sets will support all combinations. Installed fonts have the same characteristics as they would with any Windows product. For example, Monotype Corsiva is an italic font, therefore the setting of the italic font field will not have any effect on the appearance of the font.

Table D-5
Applying and Reading the Example File

Escape Sequence (primary character set)	Font Definition File Field	Font as defined in the example font definition file
ESC (1	Font Designator: 1	Use the font with a designator of 1
	Font Face: NDC Alphanumeric 1	Use the NDC Alphanumeric 1 font face. This is the default font for the designator
	Character Set: ANSI_CHARSET	Use the ANSI_CHARSET character set
	Width: 100	Use the normal width of 100 logical units
	Height: 100	Use the normal height of 100 logical units
	X Offset: 0	Use the default X axis position
	Y Offset: 6	Move 6 logical units up the Y axis
	Double Height: 0	Do not use double height font
	Bold: 0	Do not use bold font
	Italic: 1	Use italic font
	Underline: 0	Do not use underline font
	Strikeout: 0	Do not use strikeout font

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Escape Sequence (primary character set)	Font Definition File Field	Font as defined in the example font definition file
ESC (2	Font Designator: 2	Use the font with a designator of 2
	Font Face: NDC Alphanumeric 1	Use the NDC Alphanumeric 1 font face. This font file has been edited to use this font face with this designator
	Character Set: ANSI_CHARSET	Use the ANSI_CHARSET character set
	Width: 100	Use the normal width of 100 logical units
	Height: 100	Use the normal height of 100 logical units
	X Offset: 0	Use the default X axis position
	Y Offset: 6	Move 6 logical units up the Y axis
	Double Height: 0	Do not use double height font
	Bold: 0	Do not use bold font
	Italic: 0	Do not use italic font
	Underline: 0	Do not use underline font
	Strikeout: 0	Do not use strikeout font
ESC (3	Font Designator: 3	Use the font with a designator of 3
	Font Face: NDC Alphanumeric 1	Use the NDC Alphanumeric 1 font face. This font file has been edited to use this font face with this designator
	Character Set: 0	Use the ANSI_CHARSET character set. You can use either the name or the value to specify the character set. For further details, see “Character Sets.”
	Width: 100	Use the normal width of 100 logical units
	Height: 100	Use the normal height of 100 logical units
	X Offset: 0	Use the default X axis position
	Y Offset: 6	Move 6 logical units up the Y axis
	Double Height: 1	Use double height font
		Note: Blank fields can be omitted from a line as shown in this example. However, a field must be included if it is followed by other non-default fields.
	Bold:	Do not use bold font
	Italic:	Do not use italic font
	Underline:	Do not use underline font
	Strikeout:	Do not use strikeout font
ESC (6	Font Designator: 6	Use the font with a designator of 6
	Font Face: NDC Alphanumeric 1	Use the NDC Alphanumeric 1 font face. This font file has been edited to use this font face with this designator
	Character Set: ANSI_CHARSET	Use the ANSI_CHARSET character set
	Width: 100	Use the normal width of 100 logical units
	Height: 100	Use the normal height of 100 logical units
	X Offset: 0	Use the default X axis position
	Y Offset: 6	Move 6 logical units up the Y axis
	Double Height: 0	Do not use double height font
	Bold: 0	Do not use bold font
	Italic: 0	Do not use italic font
	Underline: 0	Do not use underline font
	Strikeout: 1	Use strikeout font

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Escape Sequence (primary character set)	Font Definition File Field	Font as defined in the example font definition file
ESC (7	Font Designator: 7	Use the font with a designator of 7
	Font Face: NDC Alphanumeric 1	Use the NDC Alphanumeric 1 font face. This font file has been edited to use this font face with this designator
	Character Set: ANSI_CHARSET	Use the ANSI_CHARSET character set
	Width: 100	Use the normal width of 100 logical units
	Height: 100	Use the normal height of 100 logical units
	X Offset: 0	Use the default X axis position
	Y Offset: 6	Move 6 logical units up the Y axis
	Double Height: 0	Do not use double height font
	Bold: 0	Do not use bold font
	Italic: 0	Do not use italic font
	Underline: 1	Use underline font
	Strikeout: 0	Do not use strikeout font
ESC (8	Font Designator: 8	Use the font with a designator of 8
	Font Face: NDC Alphanumeric 1	Use the NDC Alphanumeric 1 font face. This font file has been edited to use this font face with this designator
	Character Set: ANSI_CHARSET	Use the ANSI_CHARSET character set
	Width: 100	Use the normal width of 100 logical units
	Height: 100	Use the normal height of 100 logical units
	X Offset: 0	Use the default X axis position
	Y Offset: 6	Move 6 logical units up the Y axis
	Double Height: 0	Do not use double height font
	Bold: 1	Use bold font
	Italic: 0	Do not use italic font
	Underline: 0	Do not use underline font
	Strikeout: 0	Do not use strikeout font
ESC (:	Font Designator: :	Use the font with a designator of : (colon)
	Font Face: NDC Alphanumeric 1	Use the NDC Alphanumeric 1 font face. This font file has been edited to use this font face with this designator
	Character Set: ANSI_CHARSET	Use the ANSI_CHARSET character set
	Width: 100	Use the normal width of 100 logical units
	Height: 100	Use the normal height of 100 logical units
	X Offset: 0	Use the default X axis position
	Y Offset: 6	Move 6 logical units up the Y axis
	Double Height: 0	Do not use double height font
	Bold: 0	Do not use bold font
	Italic: 0	Do not use italic font
	Underline: 0	Do not use underline font
	Strikeout: 0	Do not use strikeout font

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Escape Sequence (primary character set)	Font Definition File Field	Font as defined in the example font definition file
ESC (;	Font Designator: ;	Use the font with a designator of ; (semi-colon)
	Font Face: NDC Alphanumeric 1	Use the NDC Alphanumeric 1 font face. This font file has been edited to use this font face with this designator
	Character Set: ANSI_CHARSET	Use the ANSI_CHARSET character set
	Width: 100	Use the normal width of 100 logical units
	Height: 100	Use the normal height of 100 logical units
	X Offset: 6	Move 6 logical units to the right along the X axis
	Y Offset: 0	Use the default Y axis position
	Double Height: 0	Do not use double height font
	Bold: 0	Do not use bold font
	Italic: 1	Do not use italic font
	Underline: 0	Do not use underline font
	Strikeout: 0	Do not use strikeout font
ESC (>	Font Designator: >	Use the font with a designator of > (greater than sign)
	Font Face: NDC Alphanumeric 1	Use the NDC Alphanumeric 1 font face. This font file has been edited to use this font face with this designator
	Character Set: ANSI_CHARSET	Use the ANSI_CHARSET character set
	Width: 200	Use the normal double width of 200 logical units
	Height: 100	Use the normal height of 100 logical units
	X Offset: 0	Use the default X axis position
	Y Offset: 6	Move 6 logical units up the Y axis
	Double Height: 0	Do not use double height font
	Bold: 0	Do not use bold font
	Italic: 0	Do not use italic font
	Underline: 0	Do not use underline font
	Strikeout: 0	Do not use strikeout font
ESC (?	Font Designator: ?	Use the font with a designator of ? (question mark)
	Font Face: NDC Alphanumeric 1	Use the NDC Alphanumeric 1 font face. This font file has been edited to use this font face with this designator
	Character Set: ANSI_CHARSET	Use the ANSI_CHARSET character set
	Width: 100	Use the normal width of 100 logical units
	Height: 200	Use the normal double height of 200 logical units
	X Offset: 0	Use the default X axis position
	Y Offset: 6	Move 6 logical units up the Y axis
	Double Height: 0	Do not use double height font
	Bold: 0	Do not use bold font
	Italic: 0	Do not use italic font
	Underline: 0	Do not use underline font
	Strikeout: 0	Do not use strikeout font

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Escape Sequence (primary character set)	Font Definition File Field	Font as defined in the example font definition file
ESC (@	Font Designator: @	Use the font with a designator of @ (at sign)
	Font Face: NDC Alphanumeric 1	Use the NDC Alphanumeric 1 font face. This font file has been edited to use this font face with this designator
	Character Set: GREEK_CHARSET	Use the GREEK_CHARSET character set
	Width: 100	Use the normal width of 100 logical units
	Height: 100	Use the normal height of 100 logical units
	X Offset: 0	Use the default X axis position
	Y Offset: 6	Move 6 logical units up the Y axis
	Double Height: 0	Do not use double height font
	Bold: 1	Use bold font
	Italic: 0	Do not use italic font
	Underline: 0	Do not use underline font
	Strikeout: 0	Do not use strikeout font
ESC (A	Font Designator: A	Use the font with a designator of A
	Font Face: Courier New	Use the Courier New font face. This font file has been edited to use this font face with this designator
	Character Set: ANSI_CHARSET	Use the ANSI_CHARSET character set
	Width: 100	Use the normal width of 100 logical units
	Height: 100	Use the normal height of 100 logical units
	X Offset: 0	Use the default X axis position
	Y Offset: 6	Move 6 logical units up the Y axis
	Double Height: 0	Do not use double height font
	Bold: 0	Do not use bold font
	Italic: 0	Do not use italic font
	Underline: 1	Use underline font
	Strikeout: 0	Do not use strikeout font

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Installing Fonts

Once you have obtained, created or edited the TrueType fonts to be used, they must be installed on the SST.

Use the Windows XP Control Panel to install the required fonts as follows:

1. Select **Start** | **Control Panel**
2. Select **Fonts** to display the installed fonts
3. Select **File** | **Install New Font**
4. In the **Add Fonts** dialog box, use the **Drives** and / or **Folders** options to specify the location of your fonts
5. Select the required fonts
6. Ensure the **Copy fonts to Fonts folder** check box is ticked, and select **OK**.

Your selected fonts will now be included in the *Windows \ Fonts* folder with the other installed Windows fonts.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Checking the Fonts

You can use any Windows program that uses fonts, such as WordPad, to display the fonts to prove that they are correctly installed.

If DebugLog is active, the updated font identifiers are logged.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Appendix E

Related Documentation

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Overview

This appendix lists the following documentation:

- Documentation provided with the Advance NDC CD-ROM
- Documentation that is historically linked to Advance NDC, but not provided on the Advance NDC CD-ROM
- Documentation from other sources, such as optional components and standards bodies. This documentation is not provided on the Advance NDC CD-ROM.

To read the Adobe Acrobat® Portable Document Format (PDF) documentation, you need Adobe Reader version 5.0 or later. The latest version is available free from *www.adobe.com*.

On a development system, the documentation can be installed alone or installed with the Advance NDC software. The documentation is not installed on a runtime system.

All APTRA product documentation is available under **Start | Programs | NCR APTRA | APTRA (TM) Documentation** after the relevant product has been installed.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Advance NDC Documentation

The following publications are provided on the Advance NDC CD-ROM. Printed versions can be ordered from the *NCR Publications web site*.

Note: The Advance NDC Windows help files mentioned in Appendix Table E-1 are provided as part of the APTRA Author.

Table E-1
Advance NDC Documentation

Title	Format	Description
APTRA Advance NDC, Overview B006-6597	PDF/Paper	Provides an introduction to Advance NDC, including overviews of the components contained in the Advance NDC product and provided as part of the ANDC Package on CD-ROM.
APTRA Advance NDC, Developer's Guide B006-6046	PDF/Paper	Describes how to migrate to Advance NDC. It also identifies the functionality offered by Advance NDC, and describes how to enhance it.
APTRA Advance NDC, Reference Manual B006-6180	PDF/Paper	Provides application programmers with reference information for Advance NDC, including States, Screens, and the message formats between Central and the terminal.
APTRA Advance NDC, Multi-Vendor Support Reference Manual B006-6344	PDF/Paper	Provides users with information about running the Advance NDC application on other vendors' SSTs, describing the differences between Advance NDC on NCR SSTs and other vendors' SSTs.
APTRA Advance NDC, Supervisor's Guide B006-6062	PDF/Paper	Describes the Supervisor interface, how to set up the terminal's local configuration parameters, and how to replenish the terminal.
Advance NDC Help	Windows Help	Context-sensitive help for the Advance NDC Authoring components (excluding Application Core components).
Advance NDC Application Core Help	Windows Help	Context-sensitive help for the Application Core/Supervisor Authoring components.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and/or distribution is strictly prohibited.

APTRA Author Documentation

The following titles are provided on the Advance NDC CD-ROM to support the use of the authoring environment. A printed version of the PDF file can be ordered from the *NCR Publications web site*.

Note: Advance ADE is no longer available as a separate product, but the APTRA Author is used in Advance NDC development.

Table E-2
Provided Authoring Environment Documentation

Title	Format	Description
APTRA Author, User's Guide B006-6038	PDF / Paper	Describes how to install and use the Author and Runtime Components to design, develop and maintain a self-service application.
APTRA Author Help	Windows Help	Context-sensitive help for the Author user interface and authoring errors.
Runtime Core Help	Windows Help	Context-sensitive help for the Runtime Core Authoring components and runtime errors, which are part of Advance Core Self-Service.
GUI Help	Windows Help	Context-sensitive help for the GUI Authoring components, which are part of Advance Core Self-Service.
Self-Service Core Help	Windows Help	Context-sensitive help for the Self-Service Core Authoring components, which are part of Advance Core Self-Service.
ActiveX™ Help	Windows Help	Context-sensitive help for the ActiveX™ Authoring components and runtime errors, which are part of Advance Core Self-Service.
Basic Self-Service Help	Windows Help	Context-sensitive help for the Authoring components and runtime errors provided with Advance Basic Self-Service.
ATM Help	Windows Help	Context-sensitive help for the Authoring components and runtime errors provided with Advance ATM.
Special Self-Service Help	Windows Help	Context-sensitive help for the Authoring components and runtime errors provided with Advance Special Self-Service.
Statement Printer Help	Windows Help	Context-sensitive help for the Authoring components and runtime errors provided with Advance Statement Printer.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

The PDF publications in Table E-3 below are not provided on the Advance NDC CD-ROM. These publications have not been updated, but may be useful if you are using the authoring environment to extend Advance NDC. Printed versions can be ordered from the *NCR Publications web site*.

Note: NCR recommends the use of C Exits to extend Advance NDC.

Table E-3
Additional Authoring Environment Documentation

Title	Format	Description
APTRA Advance ADE, Programmer's Guide B006-6042	PDF / Paper	Describes how application programmers should create their own C++ Worker Classes. It also contains information on using C routines to manipulate data.
APTRA Advance ADE, C++ Class Reference B006-6043	PDF / Paper	Provides application programmers with definitions for a number of C++ Runtime and Utility Worker Classes provided by NCR.
APTRA Advance ADE, Local Language Customisation Guide *B006-6037*	PDF / Paper	Describes how to customise the APTRA Advance ADE product into a local language.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

NDC+ Documentation

The publication in the following table is provided on the Advance NDC CD-ROM. A printed version can be ordered from the *NCR Publications web site*.

Table E-4
Provided NDC+ Documentation

Title	Format	Description
NDC, Using NDC Exits B006-5102	PDF / Paper	Introduces NDC Exits to experienced NDC programmers intending to develop customised terminal applications.

The publications in the following table are not provided on the Advance NDC CD-ROM. Printed versions can be ordered from the *NCR Publications web site*.

Table E-5
Additional NDC+ Documentation

Title	Format	Description
NDC, Programmer's Overview B006-2485	PDF / Paper	Provides an introduction to, and overview of, the NDC software.
NDC+, Programmer's Reference Manual B006-2486	PDF / Paper	Aimed at programmers who write host or switch applications to support NDC+ terminals, or who create the terminal configuration that customises NDC+.
NDC, Message Formats For Host Application Developers B006-4201	PDF / Paper	Designed for Central control application developers working with NDC RMX, NDCxa or NDC+. Intended to help in creating a control program that handles all the NDC variants.
NDC+, Supervisor's Reference Manual B006-2487	PDF / Paper	Designed for those people who are responsible for setting up the terminal's local configuration parameters, or for routine replenishment of the terminal.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Other NCR Documentation

The publications in the following table are provided with the software component to which they refer. Printed versions of the first two titles can be ordered from the *NCR Publications web site*.

Table E-6
Other NCR Documentation

Product	Title	Format	Description
APTRA XFS	Self-Service Support System Application User Guide B006-6167	PDF / Paper	Describes the functions that are used to perform configuration and maintenance on an SST.
	APTRA Documentation *CCM TCPIP*	HTML Help (CHM)	On-line help for the Communications Connection Manager TCP / IP module.
	APTRA Documentation *CCM PCCM*	HTML Help (CHM)	On-line help for Communications Connection Manager PCCM module.
	APTRA Communications Feature, User's Guide B006-0012	PDF / Paper	Describes how to use the Communications Feature software, which provides the files and registry settings required to run any PC Communications Module (PCCM) communications protocol on APTRA XFS.
EMV / CAM2 Exits for APTRA Advance NDC	EMV Integrated Circuit Card (ICC) Reference Manual B006-6297	PDF / Paper	Provides reference information for any APTRA Advance NDC developer who wishes to add EMV Integrated Circuit Card (ICC) Card Authentication Method (CAM) functionality to their Advance NDC application.
	APTRA Simulator	HTML Help	Context-sensitive help for the APTRA Simulator. **Note:** The XFS Simulator provided with Advance NDC is a pre-release version.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

CEN-XFS Documentation

For device access, Advance NDC supports release 3 or later of the CEN-XFS specification.

The following documents are available from the CEN site, *http://www.cenorm.be/isss/Workshop/XFS*.

Table E-7
CEN-XFS Documentation

Title	Format	Description
Extensions for Financial Services (XFS) interface specification (CWA 14050)	PDF or zipped PDF	Provides reference information for CEN-XFS specifications, release 3.00 or later.
ActiveXFS Interface Specification (CWA 13849)	PDF or zipped PDF	Provides reference information for the ActiveXFS specifications.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Appendix F

Example Custom.ini Files

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Overview

This appendix contains example *custom.ini* files for the Runtime SST and Development environments.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Runtime SST Example

The following is an example *custom.ini* file for the Runtime SST, with comment lines to explain the content:

Figure F-1
Runtime SST custom.ini example

```
; custom.ini for APTRA component customisation
;
; Registry entries have the format:
; [Registry.[configuration set name].MSI component name]
; File entries have the format:
; [Files.[configuration set name].MSI component name]
; If the configuration set name is empty, that section of the file is
; always run when the named MSI component name is installed

[Configure]
; Advance NDC has no configuration tool, so this file must be edited
; manually and this option must not be set
; Set=

; ConfigurationSet1
; registry settings set only when the MSI component 'CustRun'is
; installed and the configuration set name is ConfigurationSet1
[Registry.ConfigurationSet1.CustRun]
HKEY_LOCAL_MACHINE\Software\MyCompany\MyProduct\NumberValue1=#1
HKEY_LOCAL_MACHINE\Software\MyCompany\MyProduct\StringValue1=
MyString
HKEY_LOCAL_MACHINE\Software\MyCompany\MyProduct\MultiStringValue1=
[~]str1[~]str2
HKEY_LOCAL_MACHINE\Software\MyCompany\MyProduct\BinaryValue1=
#x010203

; files copied only when the MSI component 'CustRun' is installed and
; the configuration set name is ConfigurationSet1
; The default destination folder is [INSTALLDIR]
[Files.ConfigurationSet1.CustRun]
srcFile1.dll=c:\ssds\dll\dstFile1.dll

; ConfigurationSet2
; registry settings set only when the MSI component 'CustRun'is
; installed and the configuration set name is ConfigurationSet2
[Registry.ConfigurationSet2.CustRun]
HKEY_LOCAL_MACHINE\Software\MyCompany\MyProduct\NumberValue2=#1
HKEY_LOCAL_MACHINE\Software\MyCompany\MyProduct\StringValue2=
MyString
HKEY_LOCAL_MACHINE\Software\MyCompany\MyProduct\MultiStringValue2=
[~]str1[~]str2
HKEY_LOCAL_MACHINE\Software\MyCompany\MyProduct\BinaryValue2=
#x010203
```

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

```
; files copied only when the MSI component ‘CustRun’ is installed and
; the configuration set name is ConfigurationSet2
; The default destination folder is [INSTALLDIR]
[Files.ConfigurationSet2.CustRun]
srcFile2.dll=c:\ssds\dll\dstFile2.dll

; registry settings set every time the MSI component ‘CustRun’is
; installed
[Registry.CustRun]
HKEY_LOCAL_MACHINE\Software\MyCompany\MyProduct\InstallDir=
[INSTALLDIR]

; files copied every time the MSI component ‘CustRun’ is installed
; The default destination folder is [INSTALLDIR]
[Files.CustRun]
srcCommonFile.dll=[INSTALLDIR]MyCommonFiles\dstCommonFile1.dll
```

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Development Example

The following is an example *custom.ini* file for the development environment, with comment lines to explain the content:

Figure F-2
Development custom.ini example

```
; custom.ini for APTRA component customisation
;
; Registry entries have the format:
; [Registry.[configuration set name].MSI component name]
; File entries have the format:
; [Files.[configuration set name].MSI component name]
; If the configuration set name is empty, that section of the file is
; always run when the named MSI component name is installed

[Configure]
; Advance NDC has no configuration tool, so this file must be edited
; manually and this option must not be set
; Set=

; ConfigurationSetA
; registry settings set only when the MSI component 'CustDev'is
; installed and the configuration set name is ConfigurationSetA
[Registry.ConfigurationSetA.CustDev]
HKEY_LOCAL_MACHINE\Software\MyCompany\MyProduct\NumberValue1=#1
HKEY_LOCAL_MACHINE\Software\MyCompany\MyProduct\StringValue1=
MyString
HKEY_LOCAL_MACHINE\Software\MyCompany\MyProduct\MultiStringValue1=
[~]str1[~]str2
HKEY_LOCAL_MACHINE\Software\MyCompany\MyProduct\BinaryValue1=
#x010203

; files copied only when the MSI component 'CustDev' is installed and
; the configuration set name is ConfigurationSetA
; The default destination folder is [INSTALLDIR]
[Files.ConfigurationSetA.CustDev]
srcFile1.dll=c:\ssds\dll\dstFile1.dll

; ConfigurationSetB
; registry settings set only when the MSI component 'CustDev'is
; installed and the configuration set name is ConfigurationSetB
[Registry.ConfigurationSetB.CustDev]
HKEY_LOCAL_MACHINE\Software\MyCompany\MyProduct\NumberValue2=#1
HKEY_LOCAL_MACHINE\Software\MyCompany\MyProduct\StringValue2=
MyString
HKEY_LOCAL_MACHINE\Software\MyCompany\MyProduct\MultiStringValue2=
[~]str1[~]str2
HKEY_LOCAL_MACHINE\Software\MyCompany\MyProduct\BinaryValue2=
#x010203

; files copied only when the MSI component 'CustDev' is installed and
; the configuration set name is ConfigurationSetB
; The default destination folder is [INSTALLDIR]
[Files.ConfigurationSetB.CustDev]
srcFile2.dll=c:\ssds\dll\dstFile2.dll
```

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

```
; registry settings set every time the MSI component ‘CustDev’is
; installed
[Registry.CustDev]
HKEY_LOCAL_MACHINE\Software\MyCompany\MyProduct\InstallDir=
[INSTALLDIR]

; files copied every time the MSI component ‘CustDev’ is installed
; The default destination folder is [INSTALLDIR]
[Files.CustDev]
srcCommonFile.dll=[INSTALLDIR]MyCommonFiles\dstCommonFile1.dll
```

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Appendix G

Web Exit State

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Overview

This appendix provides information to enable a web exit.

If you require further help with web exit development, contact NCR Dundee.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Enabling Web Exits

State Table Parameters

The state table parameters, given in the Table G-1, control the behaviour of the web exit.

Table G-1
Web Exit State Table Parameters

Table Entry	Number of Characters	Contents	Description
1	1	State Type	The state type letter selected to represent the web exit state This will vary depending on the existing state type implementation The default is state type '0'
2	3	URL Index	Specifies the index into the URL table used to provide the URL from which to start the display
3	3	Browse Time	The length of time in seconds that the browser is active before Advance NDC actively takes back control A value of “000” means that no timeout is used and the HTML flow must determine a suitable time to return control to Advance NDC
4	3	Browse Complete Next State	The next state number to execute after the web page indicates that browsing is complete
5	3	Browse Timeout Next State	The next state number to execute after the time specified by the `Browse Time` state parameter expires
6	3	Browse Failed Next State	The next state number to execute when an error occurs For example, if the URL index cannot be located, or an error is found on the initial page *See* Note 1:
7		Reserved	
8		Reserved	
9		Reserved	

Note 1: The standard HTTP 404 error message is not reported as an error, and does not use the `Browse Failed Next State` parameter.

Tracking URLs

Advance NDC tracks the state of the web browser using defined URLs, described in Table G-2. When an URL in the defined list is detected, the next page is set based on the state table parameters. This mechanism is used to pass control back to Advance NDC. The

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

HTML page designer must specify these URLs as the next HTML page when appropriate.

Table G-2
Web Exit Tracking URLs

URL	Description
"http://webbrowsecomplete/"	The browse session has completed successfully Next state number is set to the entry defined in table entry 4 of the web exit state table parameters
"http://webbrowsetimeout/"	The browse session has timed out according to the input rules specified by the HTML pages Next state number is set to the entry defined in table entry 5 of the web exit state table parameters *See* Note 2:
"http://webbrowsefailed/"	An error condition has been detected during the display of web pages Next state number is set to the entry defined in table entry 6 of the web exit state table parameters

Note 2: This is not the same as the `Browse Timeout` next state. The web browse timeout is detected by the page; the `Browse Timeout` is detected by the state.

URL Index Format

The web state checks the URL index and navigates to the URL specified in that key.

For example, specifying an URL index of 1 results in the web state using the URL given in the registry for the URLIndex1 key.

The URL index uses the following registry key to define the location and name of the files to be used in local customisations:

```
HKLM\SOFTWARE\NCR\Advance NDC\Extensions\Web Exit
```

Valid values are 000 to 999. This means that you can specify URLs for URLIndex0 to URLIndex999.

Updating STCONT

To allow the use of the web exit state, STCONT has been updated so that Advance NDC recognises the new state. The STCONT file registers the state type letter and location of the implementation as follows:

```
<FS>0NssdsWebExitState<GS>SSDSNDCIMPLEMENTED<GS>0000009
9900ZZZ00ZZZ00ZZZ0000700001009990<fs>
```

In this example, the following apply:

`WebExitState` is the name of the application specified when the web exit state was built from the Author project

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and/or distribution is strictly prohibited.

SSDSNDCIMPLEMENTED indicates that the implementation of the exit is an Author flow.

If you need to change the state type letter to fit into your environment, edit the first character following the <FS>.

Updating the State Flow

The state flow must be updated so that the exit can be called at the appropriate time.

The exit is usually called from the appropriate point in the state flow. When a specific FDK is pressed, the flow exits to the web. This may require a screen update if a new selection is introduced. The state update indicates that the next state to execute is a web exit state and enables a new FDK.

To work out which state types need to be edited, trace the normal flow of the application. As highlighted in "Example Flow Trace" on page G-4, and defined in "State Table Updates" on page G-5, the following updates need to be made:

- The FDK Active Mask in state number 297 needs to be updated to enable FDK F
- The FDK F Next State number needs to be updated to point to the definition of the Web Exit State.

The Web Exit state definition also needs to be added.

Example Flow Trace

```
Applica.. 13:24:26.203 [297] [X] FDK Info Entry
Applica.. 13:24:26.203           Screen
Number                 [128]
Applica.. 13:24:26.203       Timeout Next State No       [473]
Applica.. 13:24:26.203       Cancel Next State No        [445]
Applica.. 13:24:26.218       FDK Next State No           [298]
Applica.. 13:24:26.218       Extension State No          [301]
Applica.. 13:24:26.218       Buffer Position             [030]
Applica.. 13:24:26.218       FDK Active Mask             [044]
                                          ->(60) Enable FDK F
Applica.. 13:24:26.250       Multi Language Screens      [000]
Applica.. 13:24:26.250    Extension State - Pos     [1]
Applica.. 13:24:26.250        1  2  3  4  5  6  7  8  9
Applica.. 13:24:26.250        Z0501002500000400300200l0
Applica.. 13:24:41.031
Applica.. 13:24:41.046 [298]  [W] FDK Switch
Applica.. 13:24:41.046            FDK 'A' Next State No  [000]
Applica.. 13:24:41.046            FDK 'B' Next State No  [000]
Applica.. 13:24:41.046            FDK 'C' Next State No  [205]
Applica.. 13:24:41.046            FDK 'D' Next State No  [206]
Applica.. 13:24:41.046            FDK 'F'  Next State No [000]
                                      ->(900) Execute Web Exit
Applica.. 13:24:41.046            FDK 'G' Next State No  [204]
Applica.. 13:24:41.046            FDK 'H' Next State No  [000]
Applica.. 13:24:41.046            FDK 'I' Next State No  [000]
```

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

State Table Updates

The state table needs to be updated as shown in the following:

- Enable FDK F

  297X1284734452983010306**0**000

- Goto State 900 for FDK F

  298W000000205206**900**204000000

- State 900 Web Exit State

  900090009029744506900000000

This updated state flow and any associated screens can be stored in a local customisation file to update the customisation data without impacting the host. For an example local customisation file, see “LOCAL File” on page 5-61.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Glossary

A

ABT Aggregate Builder Tool.

Active Script Host APTRA Advance worker that allows the use of additional processing to be carried out by execution of Microsoft Visual Basic Script (VBScript).

ActiveX ActiveX is the product of two Microsoft technologies called OLE and COM.

ActiveX Control A type of COM component which implements standard interfaces, and can be included in a web page and used in languages such as VBScript.

ADE Application Development Environment

ADI2 Application Device Interface 2. A proprietary interface from NCR.

Advance ADE An application development environment from NCR. Advance ADE is no longer available as a separate product, but the APTRA Author is used in Advance NDC development.

API Application Programming Interface

Application In the Author, an application is a collection of Workers that can be built to create an executable (see Self-Service Application).

Application Core The Application Core performs the SST mode handling and message processing.

APTRA Author A tool that allows you to visually design and develop a self-service application.

APTRA Component Directory Any folder containing the *_comp.ini* file.

APTRA Simulator A tool from NCR enabling the simulation of self-service devices on a development PC for the testing of applications under development.

See also **Simulated Services**.

ASCII American Standard Code for Information Interchange. A computer code for representing alphanumeric characters.

Authoring Component The smallest manageable unit in the Author. Examples of authoring components are Workers, Applications and Catalogs.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Automation Object An APTRA Advance worker which encapsulates an Automation Object, and thus enables access to these Automation Objects from within your application. Automation Objects are not displayable, but are typically data manipulation objects; for a displayable ActiveX Control, use the Active Control worker class.

Automatic INIT The process of running an electronic journal (EJ) initialisation command (INIT) automatically.

Automatic INIT EJ file The EJ file produced when running an automatic INIT.

AVI Audio Video Interleave. An audio / video standard designed by Microsoft for Windows.

B

BAPE Basic Alpha PINpad and Encryptor. A combined PIN pad and encryptor from NCR. Supports single-length DES encryption with various local and remote PIN verification schemes.

BNA Bunch Note Acceptor.

BOP Basic Operator Panel (not supported by Advance NDC).

C

Cardholder The SST customer.

Catalog A component in the APTRA Author for organising other Authoring components into manageable groups.

CCITT Consultative Committee for International Telegraph and Telephone. An international organization that develops communications standards such as X.25.

CCM Communications Connection Manager. An APTRA component providing a layer of software to separate an application from the underlying communications protocol.

CDI Common Data Interface.

CDI Store A data element that is shared by the Customisation Layer, Application Core and Supervisor. CDI stores are created and initialised by the APTRA Advance NDC runtime.

CDM Cash Dispenser Module**.** CEN-XFS class name for the Cash Dispenser service.

CDM In NDC+, Coin Dispenser Module. In Advance NDC this has been replaced by the term coin dispenser.

CEN Comité Européen de Normalisation (European Committee for Standardization). Responsible for the XFS interface specification.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Central An application that resides on a host computer or switch and interacts with Advance NDC on the SST to manage self-service transactions and maintain the SST in operation.

CHK CEN-XFS class name for the Check Readers and Scanners service class.

CLM Component Lifecycle Management.

Coin Hopper A physical container holding coins in a coin dispenser.

Coin Hopper Type An Advance NDC coin hopper type is a mapping to one logical XFS CDM cash unit. A coin hopper type is a logical representation of one or more coin hoppers containing the same type of coin.

Cold Start The first time the terminal is powered up, with no previously downloaded software.

COM Component Object Model from Microsoft. An open architecture for cross-platform development of client/server applications based on object-oriented technology. Clients have access to an object through interfaces implemented on the object (access to Methods, Properties and Events).

Component See Authoring Component and Runtime Components.

Coordinator A Worker that coordinates the work of subworkers contained in its Work Groups.

CPM Cheque Processing Module. An NCR device.

CTF Communications Template File

Customisation Layer The part of Advance NDC that performs the 'In Service' activities associated with the cardholder. For details of the Customisation Layer worker classes, refer to the on-line help in the APTRA Author.

D

DAPI1 The Basic Security firmware for an EPP.

DAPI7 The International Security firmware for an EPP.

DASH Dip And Smart Hardware. The NCR hardware used to read and write dipped/smart cards.

DCCMT Dispenser Currency Cassette Mapping Table. Customisation data sent in a message from Central and distinct from the Currency Cassettes Mapping Table held in the registry of the SST.

DebugLog A troubleshooting utility, supported from Advance NDC 3.02 onwards, that enables viewing of trace stream data in the test environment.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and/or distribution is strictly prohibited.

DEP CEN-XFS class name for the Depository Units service.

DES Data Encryption Standard. An industry-wide private key encryption mechanism. The DES algorithm uses a 56-bit private key (plus 8 bits for key integrity checking) and operates on 64-bit blocks of data.

See also: **RSA, Triple DES**

Director In Advance NDC, a worker class that controls the flow of the application through subordinate work groups of workers.

DLL Dynamic-Link Library. A library that can be linked at execution time.

DPM Document Processing Module. An NCR device.

DSM Device Status Monitor. Part of the NCR Self-Service Platform.

E

EDEP Envelope Depository service class.

EJ Electronic Journal. In Advance NDC, the electronic journal emulates the printed journal device. All the data normally recorded on the journal printer is written to the EJ log on the system disk of the SST.

EKC Encrypting Keyboard Controller. A more complex and secure version of the BAPE, providing high-level security. Not supported in Advance NDC.

EMV Europay, Mastercard and VISA. Specifications for payment systems to ensure interoperability between smart credit / debit cards and interoperability between the terminals that support them, jointly created and published by Europay International, MasterCard and Visa International in 1996.

EOP Enhanced Operator Panel, supported by Advance NDC.

EPP Encrypting PIN Pad A combined PIN pad and encryptor, supporting triple DES and RSA encryption.

EUR The ISO currency ID for the Euro.

Exits A general programming term used in Advance NDC to cover user-defined states, supervisor features, virtual controllers and special synchronisation routines called hooks. A C Exit is a DLL accessed through a specific Exit interface.

Exit State A state defined and programmed by the user to customise Advance NDC.

Exit Supervisor A supervisor function defined and programmed by the user.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

F

Fault Display This gives information to allow you to anticipate and prevent media shortages and device failures.

FDK Function Display Key. These keys are located on each side of the facia screen and enable various options to be chosen. Some screens have touch screen areas which emulate FDK functions.

Final Application In Advance NDC, an application created for delivery to the runtime system.

FIT Financial Institution Table. In Advance NDC, a FIT contains details of where and how information is stored on the magnetic strip of the card and how a transaction should be processed.

G

GBNA Global Bunch Note Acceptor.

GBP The ISO currency ID for pounds sterling.

GBRU Global Bill Recycling Unit.

GBXX GBNA and GBRU.

Global directory The user-specified directory in which Advance NDC has been installed. The default is *c:\ntglobal*.

Glyph A displayed or printed image. In typography, a glyph may be a single letter, an accent mark, or two or more typeface characters designed as a single unit (for example, œ)

GIS command A go-in-service command from Central to the SST.

GUI Graphical User Interface.

H

Hooks General term for miscellaneous Exits detailed in the MISCONT file.

HKLM HKEY_LOCAL_MACHINE. Fixed location in the registry for local configuration values belonging to the computer for any user.

HSM Host Security Module.

HTML HyperText Markup Language.

I

ID Identifier.

IDC Identification Card unit. CEN-XFS class name for the Identification Card Units service.

J

JIT Just-in-time.

JPTR CEN-XFS class name for the Journal Printer service.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

K

KVV Key Verification Value.

M

MAC Message Authentication Code.

MCI Media Control Interface. A standard interface for controlling a variety of multimedia devices and files.

MCRW Magnetic Card Reader / Writer. It reads data from a standard magnetic stripe card and, depending on the MCRW variant, writes data to one or more of the tracks on the card's magnetic stripe.

MDAC Microsoft Data Access Components. These are key technologies that enable Universal Data Access.

M-Data Maintenance Data.

MEI Media Entry / Exit Indicator.

MIDI Musical Instrument Digital Interface. Hardware specification and software protocol for communication between electronic musical instruments and computers.

MISCONT A rule file detailing miscellaneous Exits: Hooks.

Module A Module is a collection of components created by a single User. It supports the control and management of multi-developer projects.

MPEG Moving Pictures Experts Group. A set of standards for audio and video compression, or a video / audio file in the MPEG format (usually with the file extension *.mpg*).

M-Status Maintenance Status.

Multi-Vendor application An Advance NDC application capable of running on SSTs that comply with the CEN-XFS specifications.

N

NBS National Bureau for Standards**.**

NDC NCR Direct Connect. An NCR application that works in conjunction with a host- or switch-based Central application to perform self-service transactions.

NVRAM Non-Volatile Random Access Memory.

O

OEM Original Equipment Manufacturer.

OLE Object Linking and Embedding. An architecture for enabling one application to insert and access objects created in other applications.

OOS Out of Service.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

O/S Operating system, such as Microsoft Windows XP.

P

PCCM PC Communications Module. An NCR communications card allowing communications with a host.

PDF Portable Document Format. The Adobe native file format for documents viewable in Adobe Reader.

Persistence The capability of data to exist after the process or thread that created it has ceased to exist. For example, persistent data is preserved across SST shutdowns or power failures.

PIN Personal Identification Number. A secret identification number that is issued to each cardholder. Also the CEN-XFS class name for the Personal Identification Number Keypads service.

PIN pads CEN-XFS short form for the Personal Identification Number Keypads service class.

PPD Programmable Printing Depository. A type of printer supported by Advance NDC. Also called Envelope Depository.

Project In the APTRA Author, a project is a collection of components.

PTR CEN-XFS class name for the Printers service.

R

R-Data Replenishment data.

RESRVD.DEF ASCII text file containing definitions for reserved screens and keyboards, provided as part of APTRA Advance NDC. For details, refer to the *APTRA Advance NDC, Reference Manual.*

RPTR CEN-XFS class name for the Receipt Printers service.

RSA An asymmetric encryption scheme using private and public keys, named after the developers (Rivest, Shamir and Adleman) and based on the algorithm originally developed by Diffie-Hellman.

See also: **DES**

Rule File A file which tells NDC+ which Supervisor functions or States are NDC+ standard, which are user written and where the user-written routines are to be found. A rule file also gives information about the chaining of Virtual Controllers and Hooks.

Runtime Components These provide the Authoring components used to construct a self-service application and the SST runtime software.

S

S-Data Severity Data.

Self-Service Application The application that runs on an SST and processes all the necessary transactions.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Self-Service Support A 32-bit open software platform supplied by NCR for use on NCR SSTs.

Service Provider A software layer responsible for hardware abstraction providing applications with transparent access to services.

Shortcut menu A menu that appears when the user right-clicks an item. The menu lists commands pertaining with that item.

Signal Some types of worker generate a signal to allow a work flow to be created in the application.

Silent Debug A utility introduced with Advance NDC 3.02 that enables trace logging on live SSTs.

Simulated Services Simulations of SST devices. Used instead of real devices when running a Test Application.

SIU CEN-XFS class name for the Sensors And Indicators Units service.

Smart Card Common name for a card that uses an integrated circuit (microchip) rather than a magnetic stripe.

See also: ICC

SNMP Simple Network Management Protocol. A widely-used network monitoring and control protocol.

SP See **Service Provider**.

SST Self-Service Terminal.

STCONT A rule file defined by Exits when user-defined States are added.

Subworker A term given to a worker that is contained within a work group of another worker, in a worker hierarchy.

SUPCTR A rule file used by Exits when user-defined Supervisor features are added.

Supervisor The Supervisor application in Advance NDC performs the SST out-of-service functions associated with the operator/supervisor.

T

T-Code Transaction Code.

TCP/IP Transmission Control Protocol/Internet Protocol (TCP/IP) is the suite of network protocols used for all Internet traffic.

Test Application An application running on a development PC is called a test application.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and/or distribution is strictly prohibited.

TI Tamper Indicator.

Top Worker The worker at the top of the worker hierarchy.

Triple DES DES encryption performed three times successively, for greater security.
See also: **DES**

TTU CEN-XFS class name for the Text Terminal Units service.

U

UCDI User-defined Common Data Interface. Functionally identical to the CDI, it enables you to create new CDI stores (called UCDI stores).

UPS Uninterruptible Power Supply. A device that provides battery backup when mains power fails or drops to an unacceptable voltage level.

USB Universal Serial Bus. A hardware interface for attaching peripheral devices, such as disk drives.

USD The ISO currency ID for US dollars.

User A developer who has installed the Author using a personal User ID. A user can create and own components.

User Messages A director worker provided for a user to process a new message class.

User Terminal Data A director worker provided for a user to include additional data in a Terminal State message.

V

VCCONT A rule file used by Exits when Virtual Controllers are added.

VDA Vendor Dependent Application. Also CEN-XFS class name for the Vendor Dependent Application service.

VDM Vendor Dependent Mode. Also CEN-XFS class name for the Vendor Dependent Mode service.

VEROP VGA Enhanced Rear Operator Panel (not supported in Advance NDC).

Virtual Controller (or Intercept Routine) A routine defined and programmed by the user which can intercept and respond to messages.

Visual Basic Script A Microsoft scripting language which is an extension of the Microsoft Visual Basic language.

W

WAVE WAVE or *.wav* (Waveform Audio). A standard Windows based sound format.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Windows NT Microsoft Windows New Technology Workstation operating system.

Windows XP Microsoft Windows eXPerience operating system.

Work Flow In APTRA Author, a work flow allows Workers to communicate across work groups by associating two work groups. It specifies the flow of control from one work group to another.

Work Group A collection of workers with similar roles in the worker hierarchy.

Worker A graphical building block in APTRA Author. Workers are the Authoring components in a self-service application.

Worker Class Every worker belongs to a worker class. the worker class defines the characteristics and functions that any given worker will have.

Working directory In Advance NDC, the user-specified directory containing customised authored projects. Also used to build the final application.

X

XFS EXtensions for Financial Services. Application and service provider interface specifications from CEN.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Index

A

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

B

C

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and/or distribution is strictly prohibited.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and/or distribution is strictly prohibited.

D

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and/or distribution is strictly prohibited.

E

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

F

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

G

H

I

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

N

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

O

P

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

R

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

S

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

T

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and/or distribution is strictly prohibited.

U

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and/or distribution is strictly prohibited.

V

W

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

X

Confidential and proprietary information of NCR.
Unauthorised use, reproduction and / or distribution is strictly prohibited.

Cut

User Feedback Form

Title: APTRA™Advance NDC APTRA™Advance NDC,

Number: B006-6046-J000 Issue 1 **Date:** July 2007

NCR welcomes your feedback on this publication. Your comments can be of great value in helping us improve our information products.

You may send your comments to us electronically. See over for details.

Circle the numbers below that best represent your opinion of this publication.

Ease of use	5	4	3	2	1	0
Accuracy	5	4	3	2	1	0
Clarity	5	4	3	2	1	0
Completeness	5	4	3	2	1	0
Organisation	5	4	3	2	1	0
Appearance	5	4	3	2	1	0
Examples	5	4	3	2	1	0
Illustrations	5	4	3	2	1	0
Job performance	5	4	3	2	1	0
Question resolution	5	4	3	2	1	0
Overall satisfaction	5	4	3	2	1	0

5 = Excellent
4 = Good
3 = Adequate
2 = Fair
1 = Poor

0 = Not applicable

Indicate the ways you feel we could improve this publication.

- ❑ Improve the table of contents
- ❑ Improve the overview / introduction
- ❑ Improve the organisation
- ❑ Improve the index
- ❑ Make it less technical
- ❑ Make it more concise / brief
- ❑ Add more / better quick reference aids
- ❑ Add more examples
- ❑ Add more illustrations
- ❑ Add more step-by-step procedures
- ❑ Add more troubleshooting information
- ❑ Add more detail

Write any additional comments you may have below and on additional sheets, if necessary. Include page numbers where applicable.

Use the following addresses to send your comments to us electronically:

E-mail - ***userfeedback@exchange.scotland.ncr.com***
Web - ***http://www.dundee.ncr.com/infoprod/rcomment/newform/webform99.htm***

Cut

Fold

If we may contact you concerning your comments, please fill in the information below:

Name: _____

Organisation: _____

Company: _____

Address: _____

_____

Phone: _____ Fax: _____

Thank you for your evaluation of this publication. Fold the form where indicated, tape (please do not staple), and drop in the mail.

F 8763-0695

Fold

NCR

Affix Postage Stamp Here

NCR Financial Solutions Group Ltd
Information Solutions Feedback
Kingsway West
Dundee
Scotland
DD2 3XX