

iWay

iWay Summary of New Features
Version 5 Release 2.0

EDA, EDA/SQL, FIDEL, FOCCALC, FOCUS, FOCUS Fusion, FOCUS Vision, Hospital-Trac, Information Builders, the Information Builders logo, Parlay, PC/FOCUS, SmartMart, SmartMode, SNAPPack, TableTalk, WALDO, Web390, WebFOCUS and WorldMART are registered trademarks, and iWay and iWay Software are trademarks of Information Builders, Inc.

Due to the nature of this material, this document refers to numerous hardware and software products by their trademarks. In most, if not all cases, these designations are claimed as trademarks or registered trademarks by their respective companies. It is not this publisher's intent to use any of these names generically. The reader is therefore cautioned to investigate all claimed trademark rights before using any of these names other than to refer to the product described.

Copyright © 2003, by Information Builders, Inc. All rights reserved. This manual, or parts thereof, may not be reproduced in any form without the written permission of Information Builders, Inc.

Printed in the U.S.A.

Preface

This documentation describes iWay middleware solutions for tying together applications, transactions, and data on any platform.

New features for Version 5 Release 2.0 include more comprehensive platform support, enhanced connectivity, and expanded table services, among others.

In cases where a feature is not supported on certain platforms, it is stated in the feature description except for the Manageability features that are dependent on the Web Console. These, as a group, are not relevant to the MVS and VM platforms. The Table Services features are available on all platforms.

How This Manual Is Organized

This manual includes the following chapters:

Chapter		Contents
1	Data Adapter Support	Describes support for new DBMSs, new versions of existing DBMSs, and features within DBMSs.
2	OS/390 and z/OS Server	Describes enhancements to the OS/390 and z/OS Server.
3	Connectivity	Describes new features that allow applications to talk to data sources without the need to understand the underlying complexity of the transport protocols used. Also grouped with the features in this area are tools and features related to connectivity such as connection configuration and plug-ins.
4	Platform Support	Describes integration possibilities to provide access to multiple data sources.
5	Manageability and Diagnostics	Describes multiple features aimed at improving the ease of configuration and monitoring of the server environment. The release includes a new tree control for procedures.
6	Security	Describes security enhancements, which allow the use and protection of data. In Release 5 Version 2.0, iWay supports Windows NT/2000 trusted client.

Chapter		Contents
7	ETL Manager Enhancements	Describes new features that enhance the functionality of the ETL Manager
8	Maintain	Describes the enhancements that have been made to the Maintain Language and Server environment.
9	Resource Analyzer and Resource Governor	Describes new features for Data Management.
10	Table Services	Describes the enhancements that have been made to the Table Services engine.
11	SAP/IDoc Processing	Describes new features for SAP/IDOC Processing.
12	XFocus	Describes XFocus which allows you to receive, archive and process XML files. It also allows you to send an executable request contained in an XML document.

Documentation Conventions

The following conventions apply throughout this manual:

Convention	Description
THIS TYPEFACE or this typeface	Denotes syntax that you must enter exactly as shown.
<i>this typeface</i>	Represents a placeholder (or variable) in syntax for a value that you or the system must supply.
<u>underscore</u>	Indicates a default setting.
<i>this typeface</i>	Represents a placeholder (or variable) in a text paragraph, a cross-reference, or an important term. It may also indicate a button, menu item, or dialog box option you can click or select.
this typeface	Highlights a file name or command in a text paragraph that must be lowercase.
Key + Key	Indicates keys that you must press simultaneously.
{ }	Indicates two or three choices; type one of them, not the braces.

Convention	Description
[]	Indicates a group of optional parameters. None are required, but you may select one of them. Type only the parameter in the brackets, not the brackets.
	Separates mutually exclusive choices in syntax. Type one of them, not the symbol.
...	Indicates that you can enter a parameter multiple times. Type only the parameter, not the ellipsis points (...).
.	Indicates that there are (or could be) intervening or additional commands.

Related Publications

To view a current listing of our publications and to place an order, visit our World Wide Web site, <http://www.iwaysoftware.com>. You can also contact the Publications Order Department at (800) 969-4636.

Customer Support

Do you have questions about iWay?

Call Information Builders Customer Support Service (CSS) at (800) 736-6130 or (212) 736-6130. Customer Support Consultants are available Monday through Friday between 8:00 a.m. and 8:00 p.m. EST to address all your iWay questions. Information Builders consultants can also give you general guidance regarding product capabilities and documentation. Please be ready to provide your six-digit site code (xxxx.xx) when you call.

You can also access support services electronically, 24 hours a day, with InfoResponse Online. InfoResponse Online is accessible through our World Wide Web site, <http://www.informationbuilders.com>. It connects you to the tracking system and known-problem database at the Information Builders support center. Registered users can open, update, and view the status of cases in the tracking system and read descriptions of reported software issues. New users can register immediately for this service. The technical support section of www.informationbuilders.com also provides usage techniques, diagnostic tips, and answers to frequently asked questions.

To learn about the full range of available support services, ask your Information Builders representative about InfoResponse Online, or call (800) 969-INFO.

Information You Should Have

To help our consultants answer your questions most effectively, be ready to provide the following information when you call:

- Your six-digit site code (xxxx.xx).
- Your iWay Software configuration:
 - The iWay Software version and release.
 - The communications protocol (for example, TCP/IP or LU6.2), including vendor and release.
- The stored procedure (preferably with line numbers) or SQL statements being used in server access.
- The database server release level.
- The database name and release level.
- The Master File and Access File.
- The exact nature of the problem:
 - Are the results or the format incorrect? Are the text or calculations missing or misplaced?
 - The error message and return code, if applicable.
 - Is this related to any other problem?
- Has the procedure or query ever worked in its present form? Has it been changed recently? How often does the problem occur?
- What release of the operating system are you using? Has it, your security system, communications protocol, or front-end software changed?
- Is this problem reproducible? If so, how?
- Have you tried to reproduce your problem in the simplest form possible? For example, if you are having problems joining two data sources, have you tried executing a query containing just the code to access the data source?
- Do you have a trace file?
- How is the problem affecting your business? Is it halting development or production? Do you just have questions about functionality or documentation?

User Feedback

In an effort to produce effective documentation, the Documentation Services staff welcomes your opinions regarding this manual. Please use the Reader Comments form at the end of this manual to relay suggestions for improving the publication or to alert us to corrections. You can also use the Documentation Feedback form on our Web site, <http://www.iwaysoftware.com>.

Thank you, in advance, for your comments.

iWay Software Training and Professional Services

Interested in training? Our Education Department offers a wide variety of training courses for iWay Software and other Information Builders products.

For information on course descriptions, locations, and dates, or to register for classes, visit our World Wide Web site (<http://www.iwaysoftware.com>) or call (800) 969-INFO to speak to an Education Representative.

Interested in technical assistance for your implementation? Our Professional Services department provides expert design, systems architecture, implementation, and project management services for all your business integration projects. For information, visit our World Wide Web site (<http://www.iwaysoftware.com>).

Contents

1. Data Adapter Support	1-1
XA Support	1-2
ADABAS Enhancements	1-2
ADABAS Transparent Super Descriptor	1-2
ADABAS ORDER Fields	1-2
ADABAS Internal Sequence Number (ISN) Support	1-3
ADABAS Global Format Buffer ID Support	1-3
A LLBASE for HP-UX Platform	1-4
Bulk Insert Enhancements	1-5
CREATE SYNONYM From COBOL Copy Book	1-5
DB2 V7.1 FETCH FIRST n ROWS Support	1-6
ESSBASE Enhancements	1-7
UniVerse for HP-UX Platform	1-8
Enhancements for Text (TX) Fields for Relational Data Adapters	1-8
Text (TX) Field Support of 32K Text Strings	1-8
Support of WHERE Optimization Clause for Text (TX) Fields	1-9
Informix Data Type LVARCHAR Support	1-10
Enterprise Java Bean Adapter	1-11
Create Synonym Support	1-11
Access Beans Support	1-11
Micro Focus C-ISAM	1-11
MUMPS Enhancements	1-12
Progress Using ODBC	1-12
SAP R/3 Interface Enhancements	1-13
Security Support for SAP R/3 Native Interface	1-13
BAPI Support in SAP R/3 Interface	1-14
Joins Support in SAP R/3 Interface	1-14
SAP BW/BAPI	1-15
SQL Join Optimization	1-15
VARCHAR Support	1-16
DBFILE Adapter (SUFFIX=DBFILE) and DBFILE Metadata	1-16
Siebel Adapter	1-16
DB2 Table Types on OS/400	1-17
Reporting XML Content of a Data Source	1-17
Restricting size of Alpha Fields for CREATE SYNONYM and Passthru	1-18
BULL GCOS Adapter (SUFFIX=TNFOCUS)	1-18
BULL GCOS DBMS Metadata	1-19
BULL GCOS TDS/TP8 Transaction Adapter	1-19

2. OS/390 and z/OS Server	2-1
CREATE Synonym for Data Adapters	2-2
Data Adapter Support	2-2
DYNAM Allocation Support	2-3
External Sort	2-3
Hiper Memory Features	2-4
MVSAPP Feature	2-4
OS/390 and z/OS Performance Enhancements	2-4
Security Types	2-5
SMF Records	2-5
Workload Manager Enclave Integration	2-6
3. Connectivity	3-1
MQ XML Listener	3-2
MQSI Plug-in for Full XML	3-2
SOAP/XML Listener	3-4
SOAP Document Conformance	3-5
XML Document Conformance	3-5
4. Platform Support	4-1
OS/400 Operating System Support	4-2
UNIX Operating System Support	4-2
OpenVMS Operating System Support	4-2
Over 2GIG files on HP and Solaris	4-2
IBM/LINUX Access to MVS VSAM Files Using XMI Server	4-2
The XMI Client/Server Adapter	4-2
5. Manageability and Diagnostics	5-1
Connections	5-2
Agent Services	5-2
Server Control	5-4
Text Console Enhancements	5-4
Listener Statistics	5-5
LIBPATH Display for DBMS	5-5
Web Console Tree Controls for Procedures	5-5
Profiles	5-5
6. Security	6-1
Windows Authentication Methods	6-2
Windows NT/2000 Trusted Client	6-2
Administrators	6-3
HTTP Basic Authentication	6-3
SSL/TLS Support for HTTP Listener	6-3
Web Console Protected Mode	6-4

7. ETL Manager Enhancements	7-1
ETL/C Engine Enhancements	7-2
Request Generation	7-2
Application Directories	7-2
Request Logs	7-2
Web Console	7-3
Request Management	7-3
Log Management	7-3
Scheduler	7-3
Multi-Table Load	7-3
Bulk Load	7-4
Migration	7-4
Workbench	7-4
Server	7-4
Metadata Manager	7-4
Workbench Enhancements	7-5
SQL/Answer Set	7-5
8. Maintain	8-1
Active Procedure on Web Console	8-2
CONTAINS with Computed Fields	8-2
Controlling System SETs within a Procedure	8-2
Inspecting DBMS Return Codes within a Procedure	8-3
Issuing DBMS Commands within a Procedure	8-3
MAINTAIN FILETYPE Extension	8-4
New Maintain Functions	8-4
NOT_IN Logical Operator	8-4
EQ_MASK and NE_MASK Logical Operators	8-5
Performance Enhancements	8-5
Faster Execution	8-5
Smaller Memory Requirements	8-5
Shared Application Server Focus Database Access	8-5
Statement Trace	8-6
Type ON EDAPRINT	8-6
9. Resource Analyzer and Resource Governor	9-1
Resource Analyzer and Resource Governor	9-2

10. Table Services	10-1
COMBINE Command Enhancement	10-2
Compiled DEFINES	10-4
Controlling Report Page Numbering	10-6
DATASET for a Segment in a Master File	10-10
DATASET Behavior in FOCUS Data Sources	10-10
Decimal Alignment of Headings	10-13
Aligning Decimals in a Multi-Line Heading or Footing	10-17
Dynamic Data Space I/O	10-19
iWay Communications Overview	10-19
Using Data Spaces	10-20
Dynamic Data Space Prerequisites	10-20
Enabling Dynamic Data Space I/O	10-21
Dynamic Data Space I/O Startup Processing	10-26
DYNAM DSIO Usage	10-29
DYNAM DSIO Processing	10-29
Exit On Error	10-31
Controlling Batch Error Processing	10-31
Extended Currency Symbol Support	10-32
Extended Currency Symbol Format Options	10-33
FIELDTYPE=R	10-35
File Extensions for Maintain Procedures	10-38
FML Hierarchies	10-39
Requirements for FML Hierarchies	10-40
FOCSBS	10-53
Accessing a FOCUS Database Server on MVS	10-53
HNODATA	10-57
Holding Missing Values	10-61
Long Qualified Field Names	10-62
Minus Edit Format Option	10-65
Multiple FOLD-LINE in a TABLE Request	10-68
Multiple Repeated Values in FOR Statement	10-70
Multi-Variate Regress	10-72
New Functions NORMSDST and NORMSINV	10-76
Characteristics of the Normal Distribution	10-77
SET CDN Command Enhancement	10-81
SET SAVEDMASTERS	10-82
SUFFIX = COM/COMT/TABT	10-84
Accessing SUFFIX=COM Data Sources	10-85
Accessing SUFFIX=COMT Data Sources	10-85
Accessing SUFFIX=TABT Data Sources	10-85

Summary Prefix Operators	10-86
Controlling Summary Line Processing	10-91
Using Prefix Operators With Calculated Values	10-94
Using Multiple SUB-TOTAL or SUMMARIZE Commands	10-97
TABLASTPAGE	10-100
USERFCHK Setting	10-103
Wide FOCEXEC Lines	10-107
11. SAP/IDoc Processing	11-1
IDoc Listener	11-2
12. XFocus	12-1
XFocus Overview	12-2
Using XML FOCUS to Execute XML Documents	12-2
Using XML FOCUS to Archive XML Documents	12-3
Index.....	I-1

Contents

CHAPTER 1

Data Adapter Support

Version 5 Release 2.0 provides support for new DBMSs, new versions of existing DBMSs, and features within DBMSs.

For further information on these new features, see the *iWay Server Administration* manual.

Topics:

- XA Support
- ADABAS Enhancements
- A LLBASE for HP-UX Platform
- Bulk Insert Enhancements
- CREATE SYNONYM From COBOL Copy Book
- DB2 V7.1 FETCH FIRST n ROWS Support
- ESSBASE Enhancements
- UniVerse for HP-UX Platform
- Enhancements for Text (TX) Fields for Relational Data Adapters
- Informix Data Type LVARCHAR Support
- Enterprise Java Bean Adapter
- Micro Focus C-ISAM
- MUMPS Enhancements
- Progress Using ODBC
- SAP R/3 Interface Enhancements
- SAP BW/BAPI
- SQL Join Optimization
- VARCHAR Support
- DBFILE Adapter (SUFFIX=DBFILE) and DBFILE Metadata
- Siebel Adapter
- DB2 Table Types on OS/400
- Reporting XML Content of a Data Source
- Restricting size of Alpha Fields for CREATE SYNONYM and Passthru
- BULL GCOS Adapter (SUFFIX=TNFOCUS)
- BULL GCOS DBMS Metadata
- BULL GCOS TDS/TP8 Transaction Adapter

XA Support

Starting with current release, XA support has been enhanced to allow HUB/SUB servers to participate in XA compliant global transaction. XA support is extended to HUB/SUB servers and all supported associated relational adapters. The current release supports the following relational database management systems (RDBMSs):

- Oracle
- DB2 (using CLI interface)
- MS SQL Server (using OLE/DB interface)
- Informix
- Sybase ASE
- Ingress II

ADABAS Enhancements

The ADABAS Data Adapter has been enhanced to include:

- Support for Transparent Super Descriptor
- Support ORDER fields
- Internal Sequence Number (ISN) support
- Global Format Buffer ID (GFBID) support

ADABAS Transparent Super Descriptor

The ADABAS Data Adapter has been enhanced to transparently use the most efficient Super Descriptor available based on the nature of the request. A new set command, SET SYNONYM, generates the syntax required for transparent Super Descriptors in conjunction with the CREATE SYNONYM command. The CREATE SYNONYM command is used to generate the metadata (Master File) for an ADABAS file. The SET SYNONYM command can be coded in any supported server profile.

ADABAS ORDER Fields

Setting ORDERKEYS to OFF forces ADABAS interface to exclude ORDER fields from the index fields list. Order fields maintain the sequence number for each multiply occurring instance.

ADABAS Internal Sequence Number (ISN) Support

The ADABAS Data Adapter can now employ a new data retrieval strategy using Read Logical by ISN (L1) calls. You can determine the Internal Sequence Number (ISN) of a record that was read or that will be inserted into an ADABAS file.

ISN-based access is applicable only if an ISN field is described in the Master File. This field has a fieldname that is user-defined and has an ALIAS of ISN. This field can be defined only in a segments that contains non-repeating data (that is, using an access method defined in the Access File as ADBS):

```
FIELD=ISN_FIELD, ALIAS=ISN, I10, I4, $
```

The Usage format is I and the Actual format has to be I4.

Reference Usage Notes For ADABAS ISN Support

Equality tests on the ISN field can be used to retrieve a single record when:

- The report request contains an equality operator in the selection test on an ISN list.
- The ISN field is used as the cross-referenced field in the Join to an ADABAS file.

If the record defined in the test is not present in the Address Converter file, ADABAS returns the Response Code 113. The adapter returns the following message: "Record is not found".

It is also possible to assign a value for the ISN field when performing an Insert if the request contains the ISN field and the assigned value is not 0.

The adapter will issue an ADABAS N2 Direct Call to assign this ISN value to the inserted record. ADABAS returns Response Code 113 if this value was already assigned to another record in the file or if it is larger than the MAXISN in effect for the file.

You can define the Global Format Buffer ID field in the Master File manually or using the Create Synonym facility.

ADABAS Global Format Buffer ID Support

The ADABAS Data Adapter can optimize the performance of queries that use the same ADABAS field lists repeatedly. Field lists are generated in ADABAS format buffers and can be retained.

Global Format Buffer ID (GFBID) support is applicable only if a GFBID field is present in the Master File. This field has a user-defined field name, and an ALIAS of GFBID. This field is used to determine the Global Format Buffer ID that will be defined in read requests with identical field lists for the same database. The GFBID field has a Usage format of A8 and an Actual format of A8. This field can be defined only in segments that contain non-repeating data (that is, the access method defined in the Access File is ADBS).

- If the field list is changed but the same GFBID is used in a request, then incorrect results may be displayed. In some cases an error message about the possible mismatch between the Field Definition Table (FDT) and Master File will be issued.
- If two requests have the same list of selected fields but different fields are used in selection criteria, then the requests must use different GFBID values.

The GFBID field can be defined in the Master File manually or using the Create Synonym facility with option PARMS GFBID.

If a GFBID field is defined in the Master File, it can be used in a request as part of the selection test. The adapter will take the GFBID value from the selection criteria, deactivate the field, and remove it from the match array. The GFBID value is placed in the ADD5 field of the ADABAS Control Block for the request.

These values can be removed from the ADABAS queue by issuing the following adapter SET command:

```
ENGINE ADBSINX SET GFBID_OFF ALL/<value> DBID <number>
```

When a single <value> is in the SET command, all values issued for child segments are removed, as well. If <value> contains blanks, then value must be in single quotes. ALL clears all GFBID values.

A LLBASE for HP-UX Platform

The ALLBASE data source is now supported on HP-UX platform. This new data adapter supports read/write access to ALLBASE data structures as well as the CREATE SYNONYM command for the production of Master Files.

The ALLBASE Data Adapter provides the following enhanced capabilities:

- Direct SQL Passthru
- Support for the CREATE SYNONYM command
- SQL Translation
- Support for ALLBASE native data types

Bulk Insert Enhancements

Using ETL with Bulk Insert API is now supported for following types of databases:

- Sybase
- MS SQL Server

For the SYBASE Adapter, the Bulk Insert API is used in LOADONLY mode and when INSERTSIZE is greater than 1. INSERTSIZE determines how often the intermediate data flush is performed. Measurements show that intermediate flushes are necessary for optimal SYBASE server performance. As a rule of thumb the INSERTSIZE value should be from several thousand to several tens of thousand. Intermediate flushes cannot be rolled back.

For Microsoft SQL Server the Bulk API is used automatically in LOADONLY mode. Measurements show that intermediate flush does not affect performance, therefore the behavior does not depend on the INSERTSIZE value.

It has to be taken into account that for both SYBASE and Microsoft SQL Server the errors during the load, such as duplication etc., cause the batch of rows to be rejected as a whole. If it is desirable to suppress the use of Bulk Insert API use the following setting:

```
ENGINE engine SET FASTLOAD OFF
```

where *engine* is SQLMSS or SQLSYB.

This setting will suppress use of Bulk Insert API; Microsoft SQL Server adapter will resort to array insert according to the specified INSERTSIZE and SYBASE adapter will resort to individual insert operations.

CREATE SYNONYM From COBOL Copy Book

New functionality has been added to create server metadata descriptions by translation COBOL FD definitions for data adapters, including VSAM, CISAM, and Flat File. New functionality allows you to:

- Select COBOL Copy Book by name.
- Select COBOL Copy Book from a list by providing the path to the directory.
- Specify options for metadata generation, including the following:
 - On Error-continue on non-severe errors.
 - Number of Hyphens to skip-number of prefixes separated by hyphens to skip.
 - Hyphens as- replace hyphens with underscores.
 - Usage of REDEFINE-as separate segments, as comments, excludes REDEFINE.
 - Support for -Level 88, Occurs, Zoned Numeric and Numeric Fields editing options.

DB2 V7.1 FETCH FIRST n ROWS Support

Starting with the following versions of DB2, READLIMIT tests restrict the size of the answer set returned:

- 5.0 UDB (UNIX, LINUX, Windows, and OS/2)
- 5.1 OS/400 (iSeries)
- 7.1 OS/390, z/OS

In prior releases, DB2 constructed a full answer set even if a request included a READLIMIT test. The data adapter then retrieved from the answer set only the number of rows specified in the request.

When accessing a version of DB2 that supports the FETCH FIRST n ROWS clause, the data adapter translates a READLIMIT phrase to the appropriate clause to inform the RDBMS that only n rows need to be included in the answer set. For n = 1, the SQL request contains the clause FETCH FIRST 1 ROW ONLY. The reduction in answer set size reduces response time and enhances performance.

RECORDLIMIT tests do not restrict the size of the answer set returned by DB2. The full answer set is constructed and the data adapter retrieves the number of rows specified by the RECORDLIMIT test.

Note: This feature is not available in DB2 for VM or VSE.

Example Reducing Answer Set Size

The following request turns on the STMTRACE trace component and issues a retrieval request with a READLIMIT phrase. The SQL request contains the FETCH FIRST 2 ROWS clause:

```
SET TRACEUSER=ON
SET TRACEOFF=ALL
SET TRACEON=STMTRACE//CLIENT
TABLE FILE EMPINFO
PRINT LAST_NAME FIRST_NAME CURRENT_SALARY
BY EMP_ID
IF READLIMIT EQ 2
END
```

The generated SQL request contains the FETCH FIRST 2 ROWS clause:

```
SELECT T1."EID",T1."LN",T1."FN",T1."CSAL" FROM
  USER1."EMPINFO" T1 ORDER BY T1."EID" FETCH FIRST 2 ROWS ONLY;
If the test specifies READLIMIT EQ 1, the SQL request contains the clause
  FETCH FIRST 1 ROW ONLY:
SELECT T1."EID",T1."LN",T1."FN",T1."CSAL" FROM
  USER1INFO" T1 ORDER BY T1."EID" FETCH FIRST 2 ROW ONLY;
```

ESSBASE Enhancements

The ESSBASE Data Adapter has been enhanced with the following new features:

- Parent/Child view in Create Synonym.
- The SET MEASURE command which allows iWay to recognize any one dimension as an accounts/measures dimension without having to change the ESSBASE outline.
- The RESTRICTSUM command allows SUM on non-aggregated fields that appear in the access file.
- Shared Members can now be suppressed using the SET SUPSHARE command.
- ESSBASE OLAP Server's User Defined Attributes (UDA) can now be used in requests.

UniVerse for HP-UX Platform

The UniVerse data source is now supported on HP-UX platform. This new data adapter supports read/write access to UniVerse data structures as well as the CREATE SYNONYM command for the production of Master Files. The UniVerse Data Adapter provides the following enhanced capabilities:

- Direct SQL Passthru.
- Support for the CREATE SYNONYM command.
- SQL Translation.
- Support for UniVerse native data types.

Enhancements for Text (TX) Fields for Relational Data Adapters

The TX type fields have been enhanced with the following features:

- Support of text strings up to 32K.
- Support of WHERE Optimization Clause for TX Fields.

Text (TX) Field Support of 32K Text Strings

Columns described with ACTUAL=TX in a Master File now support text strings up to 32K, for example DB2 CLOB and LONG VARCHAR columns. These columns can be used for read-only operations.

Note: This feature has Candidate for Release status in iWay Version 5.20.

Syntax

How to Describe a Text Field in the Master File

FIELDNAME= *fname*, ALIAS= *sqlcolname*, USAGE=TX*nn*, ACTUAL=TX , \$

where:

fname

Is any valid field name.

sqlcolnamev

Is the RDBMS column name.

nn

Is the length of an output line for display. The maximum line length is 254 characters.

Example Describing a DB2 CLOB Column

In the Master File:

```
FIELD=CLOB_COL, ALIAS=CLOB_CO, USAGE=TX100, ACTUAL=TX, MISSING=ON, $
```

A request that retrieves this column can display text output up to 32K in length.

Support of WHERE Optimization Clause for Text (TX) Fields

When creating a table request, WHERE criteria can now reference RDBMS variable character data types such as VARCHAR, LONG VARCHAR, and CLOB (described in the Master File with USAGE=TX and ACTUAL=TX).

In addition, certain types of IF and WHERE criteria that reference variable length character data types are included in the generated SQL, causing the selection operations to be performed by the RDBMS and improving performance.

Note: This feature has Candidate for Release status in iWay 5.2.0.

The IF or WHERE test to be optimized must be a CONTAINS or OMITS test against a field described with USAGE=TX and ACTUAL=TX in the Master File. The RDBMS column must be a character variable length data type.

CONTAINS translates to LIKE in the generated SQL, and OMITS translates to NOT LIKE. The generated SQL places wildcard characters around the literal string specified in the CONTAINS or OMITS test.

Example Optimizing a Selection Test Against a Variable Length Character Column

Consider the following variation of the DB2 Master File named EMPINFO. A CLOB column has been added that contains a job description:

```
FILENAME=EMPINFO , SUFFIX=SQLDS, $
```

```
SEGNAME=EMPINFO , SEGTYPE=S0, $
```

```
FIELD=EMP_ID , ALIAS=EID , USAGE=A9 , ACTUAL=A9 , $
```

```
FIELD=LAST_NAME , ALIAS=LN , USAGE=A15 , ACTUAL=A15 , $
```

```
FIELD=FIRST_NAME , ALIAS=FN , USAGE=A10 , ACTUAL=A10 , $
```

```
FIELD=HIRE_DATE , ALIAS=HDT , USAGE=YMD , ACTUAL=DATE , $
```

```
FIELD=DEPARTMENT , ALIAS=DPT , USAGE=A10 , ACTUAL=A10 ,
```

```
MISSING=ON, $
```

```
FIELD=CURRENT_SALARY, ALIAS=CSAL , USAGE=P9.2 , ACTUAL=P4 , $
```

```
FIELD=CURR_JOBCODE , ALIAS=CJC , USAGE=A3 , ACTUAL=A3 , $
```

```
FIELD=JOBDESC , ALIAS=JDSC , USAGE=TX50 , ACTUAL=TX , $
```

```
FIELD=ED_HRS , ALIAS=OJT , USAGE=F6.2 , ACTUAL=F4 ,
```

```
MISSING=ON, $ FIELD=BONUS_PLAN , ALIAS=BONUS_PLAN , USAGE=I4 , ACTUAL=I4 , $
```

The following request specifies a CONTAINS test against the JOBDESC field:

```
SET TRACEUSER = ON
SET TRACEOFF = ALL
SET TRACEON = STMTRACE//CLIENT

TABLE FILE EMPINFO
PRINT EMP_ID LAST_NAME FIRST_NAME DEPARTMENT CURR_JOBCODE JOBDESC
WHERE JOBDESC CONTAINS 'PR'
END
```

The CONTAINS operator is translated to a LIKE operator in the generated SQL:

```
SELECT T1."EID", T1."LN", T1."FN", T1."DPT", T1."CJC", T1."JDSC"
FROM USER1."EMPINFO" T1 WHERE (T1."JDSC" LIKE '%PR%') FOR
FETCH ONLY;
```

Reference Usage Notes for Optimization of Selection Criteria Using Variable Length Data Types

The following options are not yet supported with text fields:

- CRTFORM
- TYPE
- FSCAN
- MODIFY
- HOLD
- DEFINE
- COMPUTE

Informix Data Type LVARCHAR Support

Support for Informix data type LVARCHAR has been added in this release. On the server, a LVARCHAR column is described as ALPHA.

Enterprise Java Bean Adapter

Support for Enterprise Java Beans includes Create Synonym and Access Beans.

Enterprise Java Bean Adapter currently supports access to the Entity Java Beans deployed under BEA WebLogic and IBM WebSphere Application servers.

Create Synonym Support

Create synonym uses the Interrogate Method to retrieve bean property, method, or event information. If the bean returns data, the Master File contains a field description.

Access Beans Support

Support for Enterprise Java Beans now includes Access Beans support. A table request uses the Invoke Method to access the bean. The value of the bean Primary Key is specified in the WHERE clause. Only one active Primary Key allowed in the request. The URL of the server where the bean is located will be stored in the Access File.

Micro Focus C-ISAM

The Micro Focus C-ISAM data source is now supported on RS/6000 platform. This new data adapter supports read access to Micro Focus C-ISAM data structures as well as the CREATE SYNONYM command for the production of Master Files.

The Micro Focus C-ISAM Data Adapter provides the following enhanced capabilities:

- Create Synonym (Using COBOL Copy Book)
- SQL Translation
- Support for all Micro Focus C-ISAM compression types

MUMPS Enhancements

The MUMPS Adapter is now available for OpenVMS 7.2-2 and 7.3-1 running DSM Mumps 7.3. The adapter has also been improved with the following enhancements:

- Read and write capabilities.
- Create server metadata (synonym).
- Support for transaction processing with recovery units. This allows you to protect the logical consistency of your application database during a failure.
- Support for the extended descriptions of data value for any level of subscripts.
- Support for field delimiters.
- Support for fixed and variable field redefinitions.
- Support for join between logical statements.

Progress Using ODBC

The Progress 9.1d Data Adapter has been enhanced to use an ODBC connection to the database. ODBC provides full support for ANSI SQL-92 specifications. Introduced with SQL-92 is support for BLOBs (binary large objects). Progress 9.1d Data Adapter is supported on the following platforms:

- Windows NT and Windows 2000
- IBM AIX
- Sun Solaris
- HP-UX
- Linux

Progress 9.1d Data Adapter is available on Compaq tru64 platforms only as an embedded SQL-89 interface.

SAP R/3 Interface Enhancements

The following topics describe the enhancements to the SAP R/3 interface, including the following:

- Security
- BAPI support
- Joins support

Security Support for SAP R/3 Native Interface

The Native interface for SAP R/3 has been enhanced to handle the user/password provided in

```
SQL SQLSAP SET CONNECTION_ATTRIBUTES system/user,password: 'client'
```

This feature allows you to run a secured request that involves the following:

- One or more BAPIs, as BAPIs are secured by SAP.
- SAP delivered logical databases, as the server code is secured by SAP.
- Function Modules (either SAP or customer) that are fully prototyped and that include the proper AUTHORITY-CHECK statements.

Reference Usage Notes for Security in SAP R/3 Native Interface

- If the user ID and password is not provided in the connection string, the request is executed with the user ID and password provided in sapserv.cfg for the given system.
- If user ID and password is provided in the connection string, all work except executing the request, is done with the user ID and password stored in sapserv.cfg. The user ID and password from the connection string is used only on execution of the request.

BAPI Support in SAP R/3 Interface

iWay now supports most read/only BAPIs (Business Application Programming Interfaces), including joins, as described in the following example:

```
CREATE SYNONYM baseapp/BUS0002_GETLIST
FOR BUS0002/GETLIST
BAPI DBMS SQLSAP AT I46
END
CREATE SYNONYM baseapp/BUS0002_GETDETAIL
FOR BUS0002/GETDETAIL BAPI
DBMS SQLSAP AT I46
END
JOIN BAPI0002_COMP_CODE IN COMPANYCODE_GETLIST TO
CCGD2_COMP_CODE IN COMPANYCODE_GETDETAIL
END
TABLE FILE COMPANYCODE_GETLIST
PRINT
CCGL0_TYPE NOPRINT
BAPI0002_COMP_CODE
BAPI0002_COMP_NAME
CCGD2_CURRENCY
CCGD2_LANGU
IF BAPI0002_COMP_CODE NE '2300' OR '6000'
END
```

Joins Support in SAP R/3 Interface

The Native Interface supports all from SAP /to SAP joins. However, for performance reasons, we do not recommend joining to SAP from a non-SAP data source.

It is more efficient to hold the keys in a sequential file, and then use the following code:

```
TABLE FILE SAP
PRINT FIELDS
IF KEYS IS (HOLD)
END
```

SAP BW/BAPI

SAP BW is a data adapter from SAP with consolidated information from SAP R/3 and external sources. BW is preconfigured for, but not limited to, SAP R/3 data and is based on the same business objects and metadata as R/3.

BW Reporting is based on the OLE DB for OLAP model. OLE DB is a successor to ODBC (Microsoft's Open Database Connectivity standard) designed to access multi-dimensional datastores through Multi-Dimensional Expressions (MDX).

The OLAP model defines key figures (numerics) and characteristics (alpha) to be used in multiple combinations for data analysis.

The iWay Adapter for BW offers:

- Retrieval of all BW multi-dimensional metadata objects
- SAP variable support for parameterized queries
- SAP attribute (property) support for complete reporting
- Support for SAP key and name fields
- Reporting from InfoCubes and ODS objects
- Built-in flattening for rapid conversion of datasets to rowsets
- High speed performance
- Automated Web Console for iWay metadata creation

For complete information, see the *iWay SAP BW Adapter Guide to Operations*.

SQL Join Optimization

The heterogeneous joins between two DBMSs or between the data sources on two sub-servers is enhanced in the following manner:

- Predicate of EQ and LIKE is cloned from one table to another in the join.
- Equi-joins order of retrieval is now based on the size (cardinality) of the table. Smaller tables are read first.

The cardinality may be obtained by using a new option WITH STATISTICS of the CREATE SYNONYM.

VARCHAR Support

In this release, the TX and CLOB fields will be merged to simplify the task of selecting the format.

- A char fixed length of n <= 32k. Full DML/SQL operations.
- AnV -varchar with maximum of n <= 4k. Full DML/SQL operation according to ANSI.
- TX unlimited VARCHAR (< 4k for focus DB). Limited DML operations. Limited SQL operations require field level fetch.

DBFILE Adapter (SUFFIX=DBFILE) and DBFILE Metadata

The DBFILE Adapter is used for physical, logical and multi-format logical file support using the OS/400 OPNQRYF Tool. These types of files are generally created through application languages, such as RPG and COBOL. While considered legacy types of languages, they are very actively used today.

DBFILE support has been enhanced to include adapter configuration and metadata creation from the Web Console. Metadata synonyms may also be created using the syntax:

```
CREATE SYNONYM abc FOR library/file DBMS DBFILE  
END
```

While physical and logical files can also be accessed through the DB2 Adapter, performance tests show significant performance advantages in the range of 300% (file size of 25021171 bytes, 130 fields, 37513 records) by using the DBFILE Adapter method.

Siebel Adapter

The Siebel Adapter is now available for Unix, NT and z/OS. The adapter has following capabilities:

- Read-only capability in release 5.2
- Create server metadata from Business Component and Object Component
- Support for DML and SQL syntax
- Support for multiple connections
- Full Join support
- Support for visibility mode
- Support Siebel datatypes including integer, date, time, datetime, and alpha. All others are mapped to alpha.
- Support for multivalued groups

DB2 Table Types on OS/400

Web Console Synonym Create Facility for DB2 on OS/400 has been extended beyond tables and views to include types for physical and logical files, which are also capable of being accessed by DB2.

Reporting XML Content of a Data Source

XML documents might be stored in any fields (or columns) in any data source. Reporting from such documents is supported by defining their structures as sub-trees attached to a parent segment which describes the original data source. Master File SUFFIX defines the data adapter used to read data from a data source. Segment level SEGSUF defines the root segment of an XML document representing sub-tree. The field containing the XML document is identified by POSITION keyword of a SEGMENT statement containing SEGSUF. The following is an example of a Master File describing flat file as a data source containing two different XML documents stored in each row (multipath structure):

```

FILENAME=ORDER, SUFFIX=FIX,DATASET=o1.ftm, $
SEGMENT=ROOT, SEGTYPE=S0,$
    FIELDNAME=KEYFLD, ALIAS='key',          USAGE=A10, ACTUAL=A10, $
    FIELDNAME=XMLFLD, ALIAS='xmlfld',       USAGE=A70, ACTUAL =A70,$
    FIELDNAME=XMLFLD1, ALIAS='xmlfld1',     USAGE=A70, ACTUAL =A70,$
SEGMENT=XMLORD, SEGTYPE=S0, POSITION=XMLFLD, PARENT=ROOT, SEGSUF=XML,$
    FIELDNAME=A,      ALIAS='a',           USAGE=A1,  ACTUAL=A1,  $
    FIELDNAME=B,      ALIAS='b',           USAGE=A10, ACTUAL=A10, $
    FIELDNAME=C,      ALIAS='c',           USAGE=A10, ACTUAL=A10, $
SEGMENT=D, SEGTYPE=S0, PARENT=XMLORD,$
    FIELDNAME=D,      ALIAS='d',           USAGE=A1,  ACTUAL=A1,  $
    FIELDNAME=E,      ALIAS='e',           USAGE=A10, ACTUAL=A10, $
    FIELDNAME=F,      ALIAS='f',           USAGE=A10, ACTUAL=A10, $
SEGMENT=XMLB, SEGTYPE=S0, POSITION=XMLFLD1, PARENT=ROOT, SEGSUF=XML,$
    FIELDNAME=G,      ALIAS='g',           USAGE=A1,  ACTUAL=A1,  $
    FIELDNAME=H,      ALIAS='h',           USAGE=A10, ACTUAL=A10, $
    FIELDNAME=I,      ALIAS='i',           USAGE=A10, ACTUAL=A10, $
SEGMENT=D1, SEGTYPE=S0, PARENT=XMLB,$
    FIELDNAME=J,      ALIAS='j',           USAGE=A1,  ACTUAL=A1,  $
    FIELDNAME=K,      ALIAS='k',           USAGE=A10, ACTUAL=A10, $
    FIELDNAME=L,      ALIAS='l',           USAGE=A10, ACTUAL=A10, $

```

Restricting size of Alpha Fields for CREATE SYNONYM and Passthru

To restrict the size of alpha fields created by CREATE SYNONYM command for long character fields use the following setting:

```
ENGINE engine SET CONVERSION LONGCHAR number
```

where *engine* is a valid adapter tag (suffix) and *number* is any number from 1 to 32767. This will restrict the size of An and AnV fields created by the CREATE SYNONYM command. The default value is 32767.

For full function server application the recommended values are up to 4096.

The setting also controls the size of answer set columns created by the Direct SQL Passthru.

BULL GCOS Adapter (SUFFIX=TNFOCUS)

BULL GCOS Data Adapter (SUFFIX=TNFOCUS) allows read access to UFAS and IDS2 DBMS files as well as the CREATE SYNONYM command for the production of Master Files.

UFAS and IDS2 data access and management is done by the iWay Run-Time Environment (IRTE) installed and configured on BULL GCOS systems.

The following commands allow stored procedures management and execution on BULL GCOS systems.

```
ENGINE TNFOCUS REMOTE EXEC
```

```
ENGINE TNFOCUS REMOTE PUT
```

```
ENGINE TNFOCUS REMOTE GET
```

The current version has the following limitations:

- Joins are supported via SQL access on the intermediary file.
- Multipath query is not supported.
- Aggregation and DEFINE expressions are not passed to TNFOCUS interface but are performed by iWay.

BULL GCOS DBMS Metadata

BULL GCOS Data Adapter (SUFFIX=TNFOCUS) has been enhanced to include adapter configuration and Metadata creation from the Web Console. Metadata synonyms may be created by using the syntax:

```
CREATE SYNONYM abc FOR library/file DBMS TNFOCUS  
END
```

BULL GCOS DBMS (UFAS and IDS2) metadata files must be created by using appropriate AUTOIDS2 and COBOL FD tools before running CREATE SYNONYM command.

BULL GCOS TDS/TP8 Transaction Adapter

BULL GCOS TDS and TP8 transaction monitors are accessible from iWay Transaction Application Adapter Server. BULL GCOS TDS/TP8 Transaction Adapter may be created and configured from the Web Console.

BULL GCOS TDS and TP8 transaction processing routines (TPRs) must be exposed to iWay by using Metadata creation facilities from the Web Console.

The current version supports only line mode and non interactive TPRs i.e input message (exposed TPR name and its arguments) received from iWay Connectors is directly passed on to the transaction monitor (TDS or TP8) for processing and execution answer set is returned back to the Adapter.

CHAPTER 2

OS/390 and z/OS Server

Topics:

- CREATE Synonym for Data Adapters
- Data Adapter Support
- DYNAM Allocation Support
- External Sort
- Hiper Memory Features
- MVSAPP Feature
- OS/390 and z/OS Performance Enhancements
- Security Types
- SMF Records
- Workload Manager Enclave Integration

The following topics describe enhancements to the OS/390 and z/OS Server.

CREATE Synonym for Data Adapters

CREATE SYNONYM functionality is available for the following adapters on OS/390 and z/OS:

- ADABAS
- DB2/CLI
- DB2/CAF
- FIXED
- IMS
- TERADATA
- SAP
- VSAM

Synonyms are created through the Server Console and will reside in a pre-specified directory on OS/390 and z/OS. Existing synonyms on MVS can be accessed from OS/390 and z/OS but do not have the capability to be updated.

Data Adapter Support

The following data adapters are fully supported on OS/390 and z/OS:

- ADABAS
- DATACOM
- DB2/CLI
- DB2/CAF (Call Attach Facility)
- FLAT files
- FOCUS
- IDMS/SQL
- IDMS/DB
- IMS
- MILLENNIUM
- MODEL 204
- TERADATA
- SAP
- SUPRA
- VSAM

DYNAM Allocation Support

In addition to accessing HFS-based files, Release 5.2 of the OS/390 and z/OS Server can also access files on MVS using the DYNAM command. The DYNAM command can be used to access partitioned data sets (PDSs) and data files on MVS such as FOCEXEC, MASTER, ACCESS, and DATA.

Syntax **DYNAM Command**

```
DYNAM ALLOC FILE MASTER DA qualif.MASTER.DATA SHR, REUSE
```

```
DYNAM ALLOC FILE VSAM01 DA qualif.VSAM.CLUSTER SHR, REUSE
```

```
DYNAM ALLOC FILE CAR     DA qualif.CAR.FOCUS SHR, REUSE
```

External Sort

MVS External Sort is fully supported on the OS/390 and z/OS Server. To enable the external sort, the EXTSORT setting can be placed in the edasprof.prf or in a Remote Procedure Call (RPC).

Syntax **How to Set the External Sort**

To set the external sort, the syntax is:

```
SET EXTSORT= [ON|OFF]
```

where:

ON

Enables the external sort feature.

OFF

Turns the sort feature off. This is the default.

Hiper Memory Features

Hiper Memory can be used to improve performance for temporary datasets in the Server. Hiper Memory is a file management feature that accelerates server processing that uses hiperspaces to speed up processing of temporary files. These settings are placed in the edasprof.prf file or in a Remote Procedure Call (RPC).

Syntax Using DYNAM Commands with HiperMemory

The following section describes the Hiper Memory profile commands.

DYNAM MEMIO [ON|OFF|HIPER|MEMORY]

where:

ON

Enables the HiperMemory feature.

OFF

Disables feature. I/O goes to disk. This is the default.

HIPER

Uses HIPERSPACE memory for I/O.

MEMORY

Uses main storage for I/O.

MVSAPP Feature

The MVSAPP feature allows existing applications residing on MVS to be viewed, executed and edited from the Server Console. Master, Focexec and Access PDSs allocated in the EDASTART Server JCL will be available in the Procedures and Metadata sections of the Server Console.

OS/390 and z/OS Performance Enhancements

We have seen some applications running on the OS/390 and z/OS Server with a more than 30 percent performance improvement compared to the MVS Server. This of course depends on the application, but we are confident that eventually the Server will have enhanced performance for most application scenarios.

Security Types

The OS/390 and z/OS Server use RACF, TOP SECRET and ACF2 security packages. When configuring the server in secured mode, we provide two security type settings, INTERNAL and MVS. The MVS setting prohibits the server from calling unauthorized external interfaces, while the INTERNAL setting allows calls to non-APF authorized interfaces.

You can specify this setting in the Server JCL under the EDAENV DDNAME or in the .profile on OS/390 and z/OS.

Syntax **How to Specify Security Setting**

To specify a security type, use the following

```
SECURITY_TYPE= [ INTERNAL | MVS ]
```

where:

[INTERNAL](#)

Allows the server to call non-APF authorized external interfaces for IDMS. This is the default.

[MVS](#)

Prohibits the server from calling unauthorized external interfaces.

SMF Records

The OS/390 and z/OS Server can write SMF records for each connect and disconnect. The accounting reports can be viewed through the Server Console. The recommended integer value is in a range from 128 to 255, inclusive. This number represents the SMF number used by the accounting facility when it sends records to the SMF system. Numbers from 1 to 128 are not recommended. To activate usage accounting, place the smf_recno parameter in the edaserve.cfg.

Syntax **How to Activate SMF Records**

To activate SMF records, use the following syntax.

```
smf_recno=255
```

Workload Manager Enclave Integration

A new parameter is available in the service block of the edaserve.cfg configuration file so that an enclave name can be specified. This enclave is a feature of Workload Manager (WLM).

Based on this setting, the task will join a WLM enclave when a request starts and leave it when the request finishes. This gives control as to the dispatching priority for the task over to WLM. It is now dependent on the rules for the enclave as to how the task will perform with respect to running the request.

One use of this feature can be the balancing of the workload such that a long running request will not affect a short running request. This can be achieved through WLM rules so that the long requests priority is lowered after a certain period of time. Without this feature, all requests share the regions priority.

wlm_enclave_trname = WLM service class (up to 8 alphanumeric characters)

This is a service level keyword and is used in the edaserve.cfg. It defines the WLM service class which is to be used by all requests associated with the service.

CHAPTER 3

Connectivity

Topics:

- MQ XML Listener
- MQSI Plug-in for Full XML
- SOAP/XML Listenerc

This section covers new features that allow applications to talk to data sources without the need to understand the underlying complexity of the transport protocols used. Also grouped with the features in this area are tools and features related to connectivity such as connection configuration and plug-ins.

For more information on these new features, see *iWay Connector Suite*.

MQ XML Listener

The MQ XML Listener listens on MQ for XML documents conforming to the iWay XML Document Specifications. The XFocus feature may be used to control the incoming XML from the listener to execute, execute using RPC transform, or archive.

Procedure Configuring the MQ XML Listener

The MQ XML Listener is configurable by using the HTTP Web Console.

1. Open the Communications set up page, and in the Add New Node Block section specify the following:
 - Class of Agent
 - Protocol of MQXML
 - Desired name for the block
2. Click *Continue*.
3. On the page that follows, supply qmanager, inqueue, outqueue, and errorqueue. Click *Save and restart*.
4. Click the question mark for more information.

The server is now ready to accept XML documents from MQ.

MQSI Plug-in for Full XML

The MQSI XML SupportPac provides a communication node that takes an XML document as a string from an input stream and sends this data to the iWay Server (to the SOAP Listener). This plug-in can provide an easy way to integrate the iWay Server to a user application without additional programming. The result of computation is returned to the node, which produces an XML document. This document is sent to one of three output terminals:

- RETRY
- FAILURE
- OUTPUT

Reference Usage Notes for MQSI Plug-in

- This SupportPac provides a plug-in node to be used with the IBM MQSeries Integrator Version 2.0.2 and above and iWay Server Version 5.2 and above. On Windows and UNIX, an iWay Client package is required.
- This SupportPac has been developed and tested in Microsoft Windows 2000, Sun Solaris and AIX environments.

Procedure How to Install the Plug-in Node on a Broker System

Before installing the plug-in node, complete the instructions from ISETUP in your server installation guide. To install the plug-in node, complete the following steps.

1. Start the Control Center.
2. Select the *Message Flows* view.
3. Right-click the *Message Flows* key in the left hand column.
4. Select *Add to workspace*, then select *Message Flow*.
5. Select node *MQSIXml* from the list of nodes.
6. Click *finish...*

You can now use the nodes in your message flows.

Note: We recommended you use the superuser **IBMMQSI2** to complete this task. This causes your new node to be locked under the same user ID as all the supplied IBM primitive nodes. If you do not use this user ID, the definition files in the configuration repository might be accidentally locked, and therefore open to unauthorized update.

Changing logon IDs can effect the operation of the Configuration Manager's queue manager if it is on this system, but not running as a Windows NT service.

Reference Plug-in Node Terminals

Terminal	Description
RETRY	This terminal will be selected in case server in down and no any response. Original document will be send to this queue.
FAILURE	This terminal will be selected in case something wrong in XML document. Description of the error will be send as XML document into the FAILURE terminal
OUTPUT	This terminal will be selected if transaction is completed and XML document was correctly processed. Depending on type of request data can be send to OUTPUT terminal.

Reference Usage Notes for Plug-in Nodes

- There are five plug-in node properties: Host, Port, EDACONF, Trace and PathToTrace. These attributes tell the node how to connect to iWay Server and where to locate the traces (if you need them). There are no default values for server connection. Traces turned off by default.
- In Host field, you have to specify address/name of computer where iWay server is running.
- In Port, you have to specify port which in use of SOAP Listener.
- EDACONF should point to the client directory.
- PathToTrace points to directory where do you want to place the traces.

SOAP/XML Listener

The SOAP / XML Listener listens on a TCP/IP Port number for SOAP and XML documents conforming to the iWay EDA SOAP and XML Document Specifications. The XFocus feature may be used to control the incoming XML from the listener to execute, execute through RPC transform, or archive. The SOAP/XML Listener is automatically configured for Application Adapter Server (AAS) licenses and with a port number of the server's selected TCP/IP Listener port value plus 2.

Reference Configuration Notes for the SOAP/XML Listener

- When configuring client applications for SOAP (that is, Biztalk), the only accepted transport is HTTP in the URI form of `http://host:port/cgisoap.exe`.
- When configuring the client applications for XML the URI for the client applications is of the form `host:port/cgixml.exe`. However, the iWay MQSI plug-in only requires `host:port`.
- The SOAP/XML Listener is also configurable for other server license types by using the HTTP Web Console.

SOAP Document Conformance

iWay Application Products using SOAP (that is, Biztalk) abstract the need to know the underlying conformance details. SOAP applications being written to talk directly to the SOAP/XML Listener must conform to the following.

1. Open the Communications set up page and in the Add New Node Block section select a Class of Agent, Protocol of SOAP and the desired name for the block (and click configure).
2. On the page that follows, supply the host name (usually left blank, may be set to loopback (127.0.0.1) to allow traffic only from the same machine or a specific IP when a machine is equipped with multiple network cards), service (TCP/IP Port Number), any RESTRICT_TO_IP specifications and click configure.
3. The server is now ready to accept EDA SOAP conforming SOAP documents on the specified port.

XML Document Conformance

iWay Application Products using XML (that is, MQSI Plug-in) abstract the need to know the underlying conformance details. XML applications being written to talk directly to the SOAP/XML Listener must conform to the EDA DTD. For more information on the DTD, see the *iWay Connector Suite*.

CHAPTER 4

Platform Support

Topics:

- OS/400 Operating System Support
- UNIX Operating System Support
- OpenVMS Operating System Support
- Over 2GIG files on HP and Solaris
- IBM/LINUX Access to MVS VSAM Files Using XMI Server

One of the goals of integration software is to provide access to multiple data sources where ever they exist. Version 5 Release 2.0 extends the integration possibilities to more platforms than any previous release.

For more information on these new features, see the *iWay Server for OS/390 and z/OS Configuration and Operation manual* and the *iWay Server for UNIX, Windows, OpenVMS, OS/400, and OS/390 and z/OS Configuration and Operation*.

OS/400 Operating System Support

iWay Release 5.2 now supports OS/400 V05R01 with a release directly built on V05R01 (iWay 5.1.x required use of a V04R05 build). This is in addition to V04R04 and V04R05 builds. The iWay Release 5.2 V05R01 release should be considered candidate for release when used on OS/400 V05R02 (IBM's latest OS level offering) until fully certified.

UNIX Operating System Support

iWay Release 5.2 adds support for the following UNIX Operating System and releases.

- Compaq/ALPHA Tru64 V5.1
- SUN Solaris 8
- Linux 7.2 on Intel and OS/390 platforms

OpenVMS Operating System Support

iWay Release 5.2 adds support for the following OpenVMS Operating Systems and releases:

- Compaq/ALPHA OpenVMS V7.3-1

Over 2GIG files on HP and Solaris

This feature allows the create/read of non-Focus files with size greater than 2 gigabytes (but less than 128 gigabytes) either in 32- or 64- bit mode. Supported for Solaris version 5.6.2 or greater and HP version 10.20 or greater.

IBM/LINUX Access to MVS VSAM Files Using XMI Server

In terms of functionality, this adapter consist of two parts:

- Multi-user XMI Listener running under EDAPTH Server control.
- The client, running as a part of TSCOM3 on any platform.

The XMI Client/Server Adapter

The XMI Client/Server Data Adapter provides remote read sequential access to data stored in VSAM files on MVS side.

Procedure Configuring the XMI Client/Server Adapter

To configure the XMI Client/Server Adapter, complete the following steps.

1. For the XMI Listener, modify the odin.cfg file to insert Node Listener Connection description.

```
; XMI Database
Listener
NODE = XMISU
BEGIN
  PROTOCOL = TCP
  CLASS = SUBSERVER
  SERVICE = < # of Communication port >
END
```

2. For the XMI Listener, create the file xmisprof.prf to describe all VSAM files clients are going to access. The description contains DYNAM statements where DDNAMEs are used by clients for access. For example,

```
DYNAM ALLOC DDNAME DBVSAM01 DSN DSNVSAM1 SHR REU
DYNAM ALLOC DDNAME DBVSAM02 DSN DSNVSAM2 SHR REU
.....
DYNAM ALLOC DDNAME DBVSAM0N DSN DSNVSAMN SHR REU
```

3. For the XMI Client, modify the file odin.cfg on client's side to insert Node Client Connection description (the number of the communication port for the client should be the same as for The XMI Listener).

```
; XMI Database Client
NODE = XMISU01
BEGIN
  PROTOCOL = TCP
  CLASS = SUCLIENT
  HOST = <Name of Host >
  SERVICE = < # of Communication port >
END
```

Example Using the XMI Client/Server Adapter

After the XMI Listener is running, the client can perform the following request.

```
USE DBVSAM02 on XMISU01
TABLE FILE DBVSAM02
PRINT .....
END
```

CHAPTER 5

Manageability and Diagnostics

Topics:

- Connections
- Agent Services
- Server Control
- Text Console Enhancements
- Listener Statistics
- LIBPATH Display for DBMS
- Web Console Tree Controls for Procedures
- Profiles

Version 5 Release 2.0 provides multiple features aimed at improving the ease of configuration and monitoring of the server environment. The release includes a new tree control for procedures.

For more information, see *iWay Server for UNIX, Windows, OpenVMS, OS/400 and OS/390 and z/OS Configuration and Operation*.

Connections

Manageability of a server has been enhanced by consolidating the concept of connections as separate objects which can be managed through the console. A connection refers to a physical connection between Client and Server. In the previous release only queued connections were materialized on the console. Now the connections view shows both active and queued connections and the queue view has been removed.

There are two types of connections, active and queued:

- An active connection is assigned to a session in a data agent.
- If there are no agents available for a service and the service is configured with a queue, the connection remains queued.

A queued connection waiting for an agent becomes active as soon as an agent is available. If the maximum time to wait in the queue is reached, the connection is cancelled.

You can see the statistics of any connection, either active or queued, from the Connections Panel on the Web Console. The connection can also be cancelled from this panel.

Agent Services

Manageability has been enhanced by providing the ability in a configuration to divide its agents into different groups called services. Each agent is running for a specific service. Each service can have different values for some of the configuration parameters. Services are configured in `edaserve.cfg` configuration file. At least two services must be configured in `edaserve.cfg` with names "DEFAULT" and "WC_DEFAULT". "WC_DEFAULT" service is to serve administration requests from Web Console only. More services can be optionally configured as required.

To manually define a service in the edaserve.cfg, use the following format:

```
SERVICE = <service_name>
BEGIN
  deployment = <private|pooled>
  eda_user_fixed = <yes|no>
  pooled_user = <pooled_user_id>
  pooled_password = <pooled_user_password>
  maximum = <number>
  number_ready = <number>
  maximum_q = <number>
  queue_limit = <number>
  idle_session_limit = <number>
  idle_agent_limit = <number>
  cpu_limit = <number>
  memory_limit = <number>
  max_sessions_per_agent = <number>
  profile = <file.fex>
  max_connections_per_user = <number>
END
```

There are two new keywords in the service block, profile and max_connections_per_user. profile is used to specify a profile name for execution during startup of the agents for this service. By default the Baseapp directory is searched for the profile. Other Application directories can be specified using the format APP/file.fex. max_connections_per_user is used to limit the number of concurrent connections a user can have. The number of concurrent connections allowed for a user can be up to the number specified by max_connections_per_user. If max_connections_per_user is not set or set to -1 then the number of concurrent connections is unlimited (this is the default behavior). All other parameters have the same meanings and relationships as they used to have for the whole server, except now in the scope of a service.

To connect to an agent of a specific service, use the following format in odin.cfg for the Client:

```
; TCP Loopback Client
NODE = LOOPBACK
BEGIN
  PROTOCOL = ...
  CLASS = CLIENT(<service_name>)
  HOST = ...
  SERVICE = ...
END
```

If there is no specific <service_name> specified then by default the client will be connected to an agent from the DEFAULT service.

For more details see the Configuration manual.

Server Control

Manageability has been enhanced by providing a new server control which can be helpful to make application changes on a running production server.

This new control allows the administrator to block new connections from non administrator users while configuration work is done on the server:

- From the Web Console, it is available in the Workspace submenu as Quiesce or Enable
- From the text console, use `-quiesce newconnect` to disable new connections and `-enable newconnect` to enable them again

When new connections are disabled, they are rejected with a resource limit error (-33) unless the connecting user is a Server Administrator. A Server administrator is still allowed new connections because some configuration changes from the web console require an agent connection.

The administrator is still responsible for looking after the existing sessions: wait for them to finish before disrupting them (e.g. stopping old agents or restarting server after configuration changes) and/or manually terminate selected sessions. The quiesced/enabled state is also preserved in `edaserve.cfg` under keyword `quiesce_connections` so that the server can stay quiesced across one or more restarts during configuration changes.

Text Console Enhancements

The Text Console (invoked by starting the Workspace Manager from the command line using `edastart` with no options) can now be exited and later reentered on any system without stopping the Workspace Manager.

The `-quit` option is now recognized from within the text console and will return to the shell, suspending the copying of `edaprint` log output that was being sent to the terminal under the text console, but without stopping the server nor the output to the `edaprint` log file itself.

The `edastart -console` option can be used to later return to the text console and resume monitoring of future `edaprint` messages output to the terminal under the console. This command can be used even if there was no text console to begin with (e.g. Workspace started with `-start` option). It cannot be used if there is already a text console running (there can be only one).

The description of all `edastart` options available in this release can be accessed by issuing `edastart -?`.

Listener Statistics

Listener statistics can be displayed on Web Console by selecting appropriate listener's Process ID in list of listeners.

LIBPATH Display for DBMS

When configuring a DBMS adapter on UNIX environments the Shared Library Path is now displayed. This is displayed on the Adapter ADD panel.

Web Console Tree Controls for Procedures

The Procedures page of the Web Console has been enhanced to support a directory tree structure. Directories on the Application Path are displayed in a tree format with procedures or HTML files appearing in individual folders. Clicking on a procedure displays multiple options that are available to be performed with that procedure. Selecting an HTML file from the Procedures page opens that file.

Profiles

Manageability of a server has been enhance with the ability to specify the location for user level profiles. A profiles directory is created during installation and pointed to by a new keyword, `edaprfu`, in the `edaserve.cfg`. By default it is under the `ibi` directory, i.e. `c:\ibi\profiles` in windows environments. This location can be overwritten by setting an environment variable `EDAPRFU` or via the keyword `edaprfu` in the `edaserve.cfg`. When a user connects to the server, the location pointed to by the environment variable `EDAPRFU` is searched. If these variables are not set or the user profile is not found in those directories, then the location pointed to by the keyword `edaprfu` in the `edaserve.cfg` is searched.

For the Server level profile, `edasprof.prf`, if the environmental variable `EDAPROF` is set this directory is searched first. If the `edasprof.prf` is not found, then `EDACONF/etc` will be searched.

CHAPTER 6

Security

Topics:

- Windows Authentication Methods
- Windows NT/2000 Trusted Client
- Administrators
- HTTP Basic Authentication
- SSL/TLS Support for HTTP Listener
- Web Console Protected Mode

This section covers security enhancements, which allow the use and protection of data. In Release 5 Version 2.0, iWay supports Windows NT/2000 trusted client.

For more information on these new features, see the *iWay Server Administration* manual.

Windows Authentication Methods

Security on Windows NT/2000 has been enhanced to support different Microsoft Authentication methods. This is done with a new keyword in the server configuration file (edaserve.cfg). These methods correspond to users rights within Microsoft security. A method indicates the mode used to authenticate and impersonate the client at the win32 level. Windows only allows the access for users that have the corresponding user right.

```
logon_method = interactive [/batch/network]
by default, logon_method = interactive
```

`interactive`

This corresponds to the user right *log on locally*.

`network`

This corresponds to the user right *access this computer from the network*.

`batch`

This corresponds to the user right *log on as a batch job*.

If the user right for *log on locally* is disabled, but *access this computer from the network* is enabled then the `logon_method` should be changed set to `network`.

Windows NT/2000 Trusted Client

Functionality of the server has been enhanced to support a trusted Windows NT/2000 client. This is optional and can take effect only on platforms where a password is not required. Since Windows requires a password to impersonate a user, this is not available on Windows iWay Servers. This new option can be accomplished through a new keyword in server configuration file (edaserve.cfg). On MVS this keyword is set in the service section of the server configuration file.

Syntax How to Specify a Trusted Windows NT/2000 Client

Syntax is:

```
trust_nt = [y|n]
```

where:

`y`

Specifies a trusted Windows client.

`n`

Does not allow a trusted client. This is the default.

Administrators

Functionality of the server has been enhanced to support a list of administrators. This list can be defined using the `server_admin_id` keyword in the server configuration file (`edaserve.cfg`). Each member of the list includes user id, password, and administration level information. Users defined in the list may have Server or Application administration level. A Server administrator can perform all the administrative tasks available through Web Console operations, including the creation of more Server or Application administrators. If there is more than one Server Administrator defined, the first valid member of the list will be used for impersonation of FDS and other special services. An Application Administrator is limited to administrative tasks without the ability to add/change/remove many parameters. All other users (not in the list) are limited to monitoring functions only. For more information see the Administration documentation and the configuration keyword reference.

HTTP Basic Authentication

In addition to the IWA and EDA Basic Authentication methods, HTTP listener now supports HTTP Basic Authentication. This new authentication method can be activated by setting `SECURITY` keyword in HTTP listener node block in ODIN communication configuration file.

```
NODE = LST_HTTP
BEGIN
  PROTOCOL = HTTP
  SERVICE = 8101
  CLASS = AGENT
  SECURITY = HTTPBASIC
END
```

This authentication method uses a Web Server logon options. We do not recommend using this method as it is not encrypted.

SSL/TLS Support for HTTP Listener

HTTP Listener has been enhanced to support SSL (Secure Socket Layer) version 3 and TLS (Transport Layer Security) version 1 protocols through OpenSSL. Listener supports DH and RSA algorithms for key exchange, DSS and RSA for authentication, 3DES and RC4 for encryption/decryption, MD5 and SHA1 for hashing. To enable the SSL/TLS protocol, a certificate and a private key files must be specified in `SSL_CERTIFICATE` and `SSL_PRIVATE_KEY` in `odin.cfg` file. Listener will present the certificate to clients, and clients will verify the certificate to determine if the listener can be trusted. Optionally, listener can authenticate clients by providing trusted CA certificates in `SSL_CA_CERTIFICATE`. Only clients with certificates signed by one of the trusted CAs will be established a secure connection to the listener.

Web Console Protected Mode

A third security mode called WCPROTECT has been added. This mode allows to run the server without operating system security while still being able to protect access to the administrative functions of the Web Console. For more details see the Security Overview section of the documentation.

CHAPTER 7

ETL Manager Enhancements

Topics:

- ETL/C Engine Enhancements
- Request Generation
- Application Directories
- Request Logs
- Web Console
- Scheduler
- Multi-Table Load
- Bulk Load
- Metadata Manager
- Workbench Enhancements

This section covers new features that enhance the functionality of the ETL Manager.

For further information on these new features, see *iWay ETL Manager*.

ETL/C Engine Enhancements

Request Generation

ETL requests are now generated at the time they are saved as stored procedures (focexecs). This enhances the functionality of the ETL Engine in the following ways:

- Overhead required to run a request is greatly reduced. Smaller requests especially benefit from this with dramatically reduced run times.
- Problem determination is simplified because requests can be run from the Interactive Agent Tool (edastart t).
- You can edit requests to add functionality or to address problems.
- Saving and opening requests is much faster.
- Resource Analyzer can monitor requests by name.
- Configuration is simplified because a relational or FOCUS database is no longer required for storing requests. A database is still used for the log and statistics tables.
- Request names can be up to 32 characters, allowing for more descriptive names.
- A Userid, previously limited to eight characters, can now be up to 32 characters long, increased from 8.

Application Directories

The grouping of ETL requests formerly known as *Categories* has been enhanced and renamed. The request browser displays folders for each application. The folders can be collapsed to show only the application name or expanded to see all requests stored in that application.

Requests are stored on the server in the specified application directory.

The server or user profile Application Path setting determines which application directories a user can access for both ETL requests and synonyms.

Request Logs

The request logs have been simplified and are easier to understand. Parameters for stored procedures appear in the log in addition to procedure names.

Output from user written stored procedures (-TYPE commands) are captured by the logs.

Web Console

ETL requests can now be managed from the iWay Web Console. While the ETL Workbench is still used to build and edit requests, most management activities can now be performed from the Web Console.

Request Management

The Web Console displays a list of all ETL requests stored on the server. A request can be selected to view, run or run deferred. The most recent log and all logs for a request can also be viewed.

Log Management

The Log and Statistics tables can be viewed, created or cleaned from the Web console. The Statistics report, which shows one line for each run of a request, is color-coded to indicate requests that are in progress, successfully completed, or failed. Each report also has a link to the detailed log for that run of the request.

When viewing the Statistics table, the job ID is a link that can be used to view the log for that job.

Scheduler

Scheduled requests are initiated using the deferred receipt server; the same technology used by majority of WebFOCUS server installations.

Requests can now be scheduled to the minute, rather than the fifteen-minute period supported by previous releases. Requests can be scheduled to run as often as every five minutes, rather than the hourly interval in previous releases.

The ETL scheduler can be accessed through the Web Console.

Multi-Table Load

Multiple target tables, up to 15, can now be loaded from a single extract. The tables can be in different data sources or different servers within one data source. For example, you can load an MS SQL Server table and an ORACLE table with a single extract. Multiple flat files can also be created with a single extract.

Bulk Load

A new bulk load capability is incorporated into the ETL server that generates bulk load commands.

New in this release is support for Sybase Adaptive Server IQ. Improved support for transformations to "flat file" targets. Transformations can be applied in a single pass of the source data. Mapping to a pre-defined flat file target structure can be used.

Migration

Enhancements have been made to the ETL migration features in the following areas:

- Workbench
- Server

Workbench

As with previous releases requests can be migrated from the ETL workbench. Copy-and-Paste or Drag-and-Drop can be used to migrate requests from one server to another.

Server

A server procedure, initiated from the Web Console, is provided that migrates requests from a 4.3 or 5.1 server to the new format to facilitate migration of large numbers of requests.

Metadata Manager

The Metadata Manager has been enhanced in the following ways:

- Previous versions of ETL requests and table metadata are stored and can be displayed
- Versions of ETL requests can be compared and the differences displayed
- Previous versions of ETL requests can be imported to the ETL Manager workbench
- A graphical File Tool supports editing file descriptions

Workbench Enhancements

Enhancements to the Workbench include the following features:

- Column lists now identify key columns not only for local tables but also for remote tables.
- Indexed columns in FOCUS files are now identified.
- Copy/Cut and Paste can be used in all grids to copy or move column names and transformations from one part of a request to another and between requests.
- Self-referential transforms (i.e.: CNTR = CNTR + 1) are no longer flagged as errors.
- An open request can be checked out to a specific user preventing other users from editing it until it has been checked in.

- The TAB key can be used to move between entry areas.
- Scroll bars are available in the workflow.

SQL/Answer Set

The SQL/Answer set object supports user-written FOCUS requests. The SQL/Answer object now allows you to move SQL SELECT statements between applications by copying and pasting.

Source

The LAST function can be used in an extract transform allowing comparison of the current and previous value of an input column. A new Test button allows testing extract transforms.

Target

For new relational database target tables a new pull-down provides a list of all connections which may be servers or data sources.

A new Validate option allows validation of input rows as they are loaded into the target tables and sending rejected rows to a log file.

For flat file targets mapping to an existing file description is now supported greatly simplifying creating files in a specific pre-defined layout for application loading. New support for MODIFY to a flat file is used.

Load

For existing relational database target tables a new **Prior to load** parameter allows selection of Leave or Delete table prior to load and a new option in this release Truncate which removes rows from a table more quickly since no logging is performed. For clarity, for a FUSION or FOCUS target the Prior to load options are Leave or Drop file.

Transport Simplification

A new, optional, transport object simplifies specification of FTP parameters.

Nesting Dependent Requests

Dependent requests can now be nested, for example request A can have request B as a dependency, request B can have request C as a dependency.

E-mail Request Notification

You can now specify in the Workbench that an e-mail message be sent when a request starts, completes or fails.

Optimized Load

Loading relational database targets is much faster when optimized load capabilities are used to insert a block of rows at once. For MS SQL Server and Sybase, Fastload is used; for other relational databases, Array Insert is used. Optimized load is available for relational database targets on Windows and UNIX. To utilize optimized load for these platforms, be sure that *Include duplicates* is selected on the load object so that no pre-processing is performed.

A new user entry area labeled *Optimize Load* accepts the number of rows to load at once as input.

- If the number one (1) is entered (the default), row-at-a-time insert is used and the behavior is the same as in previous releases.
- If a number greater than one (1) is entered, optimized load is used. Note that when optimized load is used:
 - Loading is much faster since a block of rows are inserted at once. In iWay Testing, up to six times faster loads have been observed.
 - Input data should be clean. If any one row in the block causes a database constraint violation such as not null or unique index, the entire block is rejected.
 - No row counts are available for the number of records inserted or rejected in the detail log or statistics. N/A (not available) will appear in the log instead.
 - Record logging (used to write rejected records to a flat file for review) is not available

CHAPTER 8

Maintain

Topics:

- Active Procedure on Web Console
- CONTAINS with Computed Fields
- Controlling System SETs within a Procedure
- Inspecting DBMS Return Codes within a Procedure
- Issuing DBMS Commands within a Procedure
- MAINTAIN FILETYPE Extension
- New Maintain Functions
- NOT_IN Logical Operator
- EQ_MASK and NE_MASK Logical Operators
- Performance Enhancements
- Shared Application Server Focus Database Access
- Statement Trace
- Type ON EDAPRINT

This section covers the enhancements that have been made to the Maintain Language and Server environment.

Active Procedure on Web Console

The currently running Maintain procedure is displayed on the Web Console in the Last Command column in the agent and session screens.

CONTAINS with Computed Fields

A declared/computed field can now be used as a CONTAINS or OMITTS value in a NEXT command against a database.

Example Using CONTAINS with a Computed Field

```
Compute namepart/a0=searchname;
```

```
For All Next Employee Where Empname CONTAINS namepart;
```

Controlling System SETs within a Procedure

To invoke a system set, use the SYS_MGR.FOCSET function to pass the name of the SET option and the value. The following options may be SET:

- EMGSRV
- MESSAGE
- WARNING
- TRACEON
- TRACEOFF
- TRACEUSER
- CDN
- COMMIT
- DEFCENT
- YRTHRESH
- DATEDISPLAY
- NODATA
- LANGUAGE
- USER
- PASS

Example Invoking a System Set

```
SYS_MGR.FOCSET("CDN", "ON");
```

Inspecting DBMS Return Codes within a Procedure

The new DBMS_ERRORCODE property of the SYS_MGR object has the SQLCODE of the last database command. Applications can use this to gather more details of why an operation failed then is available in FOCERROR. For example, an INCLUDE might fail due to referential integrity failure or lack of permissions or any number of reasons. DBMS_ERRORCODE will have the specific SQLCODE from the DBMS so your application can take the correct action.

Reference Usage Notes for DBMS Return Codes

Different DBMS vendors use different SQLCODEs. Some vendors have even changed their SQLCODEs between releases. To insure easy portability in the future, use a coding technique that is easy to adjust.

```
Compute Ref_Integ_Code/i8 = -843;
...
Include ProdOrder from OrdStk;
If SYS_MGR.DBMS_ERRORCODE eq Ref_Integ_Code Then ...
```

Better yet, put the Computed field in a common procedure that is IMPORTed.

Issuing DBMS Commands within a Procedure

All of the direct pass through commands and access parameters can be issued within a Maintain procedure by using the ENGINE function of the SYS_MGR object. Applications can create and drop tables, create indexes, and even run RPCs. The ENGINE function is passed the engine name and the command and returns a return code.

Example Issuing a DBMS Command with Maintain

```
iret=SYS_MGR.ENGINE("DB2PROD", "DROP TABLE LOGACT");
```

MAINTAIN FILETYPE Extension

Beginning with this release, Maintain procedures have a new file extension. For portable platforms such as NT and Unix, the new extension is .mnt. For both MVS and CMS, the extension is MAINTAIN. For the portable platforms, front-end tools will generate .mnt as the extension for the Maintain procedure. On MVS you must allocate the ddname MAINTAIN; on VM the file type is MAINTAIN.

Maintain commands such as MNTCON EX or MNTCON COMPILE will first look for a procedure with the MAINTAIN or .mnt extension. If Maintain does not find the procedure with the new extension it will look for the FOCEXEC on Mainframe platforms, or .fex on portable platforms.

New Maintain Functions

The following functions are new to the 5.2 release.

- CHAR2INT returns the integer value of an ASCII or EBCDIC character.

```
int=char2int("x");
```
- INT2CHAR returns a character from the current 256 element code-page.

```
char=int2char(93);
```
- NLSCHR converts characters from the native English code page to the running code page. For example, if you want to display a dollar sign, no matter what the code page is,

```
compute adollar/a1=nlschr("$");
```

This is most useful when hosting Web applications on EBCDIC host with non English code pages.

NOT_IN Logical Operator

The new NOT_IN operator is true when a field is not in a list of values.

Example Using the NOT_IN Operator

```
For All Next Country Where Country NOT_IN ("ENGLAND", "FRANCE");
```

EQ_MASK and NE_MASK Logical Operators

The new EQ_MASK and NE_MASK operators allow you to compare a database field to a computed field that uses \$ as a wild card character.

Example Using EQ_MASK Operator

```
Compute afld/a8="$$$LAND";
```

```
For All Next Country Where Country EQ_MASK afld;
```

Performance Enhancements

Performance enhancements include execution and memory enhancements.

Faster Execution

Execution time for Web transaction screens has been cut by 50%.

Smaller Memory Requirements

Memory usage has been optimized in a number of areas:

- Compiled procedures take 30% less memory on average.
- User stacks start with a smaller initial size.
- WEBFORMS use less memory.

Shared Application Server Focus Database Access

Accessing Focus databases when the server is running in SAS. mode (MNTCON STARTSERVER) is now supported.

Statement Trace

New trace controls allow application developers to trace the logic flow of Maintain procedures as well as measure CPU and memory used at the procedure, case, and statement level. Procedures must first be compiled to provide the desired level of tracing.

- MNTCON COMPILE procname will produce trace op-codes for each CASE entry, ENDCASE, CALL and EXEC statement. (Note: A MNTCON EX procname does a compile behind the scenes and runs the procedure without writing the compiled version to disk; it will therefore also supply trace information at runtime if the SET TRACEON options have been set on the server, as outlined below.)
- MNTCON DEBUG COMPILE procname will produce trace op-codes for every Maintain statement, such as COMPUTE, IF, FOR ALL NEXT.

Syntax **How to Set Statement Trace**

Tracing must first be enabled on the server.

- SET TRACEON=MNTSTMT produces basic Maintain trace information such as filename, casename, line number and milliseconds
- SET TRACEON=MNTPERF used in addition to SET TRACEON=MNTSTMT causes tracing to include memory and CPU performance statistics.

The trace output can be viewed from the server's Web Console or by developing reports using the MNTSTMT master file, which describes the fields used to format the trace information.

Type ON EDAPRINT

Issuing a TYPE ON EDAPRINT ; directs output to the edaprint.log file which is viewable from the Web Console or SDSF display on MVS. Use this in production applications for communicating information to the application administrator.

Example **Using Type ON EDAPRINT**

```
type on EDAPRINT "this will appear in the edaprint log"
```

CHAPTER 9

Resource Analyzer and Resource Governor

Topics:

- Resource Analyzer and Resource Governor

This section covers new features for Data Management.

Resource Analyzer and Resource Governor

Resource Analyzer and Resource Governor provide resource monitoring to help identify bottlenecks and inefficiencies in your enterprise, along with intelligent preemptive query governing to halt excessive queries before they bog down critical systems.

The following are new enhancements for Resource Analyzer and Resource Governor:

- Configuration of the Resource Analyzer and Resource Governor internal tables is now available to Server Administrators from the Web Console. The configuration process prompts you for the information required to create these tables. You can also turn ON/OFF Global Request Monitoring and Global Stored Procedure (focexec) monitoring when the internal tables are created.
- Resource Analyzer and Resource Governor reports are now accessible from the Web Console using the HTTP Listener.
- Resource Analyzer and Resource Governor now monitor long procedure names for WebFOCUS applications. In addition, Resource Analyzer and Resource Governor now store up to 66 bytes of a stored procedure name and all reports support the new length.
- New reports have been added including Long-Running Requests, Large Volume Requests and Candidates for Column Index.

CHAPTER 10

Table Services

This section covers the enhancements that have been made to the Table Services engine.

Topics:

- COMBINE Command Enhancement
- Compiled DEFINES
- Controlling Report Page Numbering
- DATASET for a Segment in a Master File
- Decimal Alignment of Headings
- Dynamic Data Space I/O
- Exit On Error
- Extended Currency Symbol Support
- FIELDTYPE=R
- File Extensions for Maintain Procedures
- FML Hierarchies
- FOC SBS
- HNODEATA
- Holding Missing Values
- Long Qualified Field Names
- Minus Edit Format Option
- Multiple FOLD-LINE in a TABLE Request
- Multiple Repeated Values in FOR Statement
- Multi-Variate Regress
- New Functions NORMSDST and NORMSINV
- SET CDN Command Enhancement
- SET SAVEDMASTERS
- SUFFIX = COM/COMT/TABT
- Accessing SUFFIX=COM Data Sources
- Summary Prefix Operators
- TABLASTPAGE
- USERFCHK Setting
- Wide FOCEXEC Lines

COMBINE Command Enhancement

You can now use the COMBINE command to access up to 63 data sources in a single MODIFY request. The COMBINE command creates one logical structure from physically separate data sources. Each COMBINE structure can include up to 64 segments, the first of which is the virtual root segment created by the COMBINE command. The COMBINE structure lasts for the duration of the FOCUS session.

Reference Support for the COMBINE Command

The COMBINE command is supported for the following types of data sources:

- FOCUS
- XFOCUS
- Fusion
- Relational
- ADABAS

While data sources in a COMBINE structure can have different suffixes, you can only combine a FOCUS, XFOCUS, or Fusion data source to data sources of the same type (although FOCUS and XFOCUS data sources can be referenced in the same COMBINE). The combined structure can include 63 data sources each consisting of a single segment, or a combination of data sources whose segments total to 63 (the addition of the root makes the total 64).

The ? COMBINE query shows up to 63 entries and the CHECK FILE command shows up to 64 segments and files.

Syntax How to Create a COMBINE Structure

```
COMBINE FILES file1 [PREFIX pref1|TAG tag1] [AND]
.
.
.
filen [PREFIX prefn|TAG tagn] AS asname
```

where:

file1 ... *filen*

Are the Master File names for the data sources you want to modify. You can specify up to 63 names.

pref1 ... *prefn*

Are prefix strings for each data source; up to four characters. They provide uniqueness for field names. You cannot mix TAG and PREFIX in a COMBINE structure.

tag1 ... *tagn*

Are aliases for the Master File names; up to eight characters. FOCUS uses the tag name as the qualifier for fields that refer to that data source in the combined structure. You cannot mix TAG and PREFIX in a COMBINE, and you can only use TAG if FIELDNAME is set to NEW or NOTRUNC.

AND

Is an optional word to enhance readability.

asname

Is the required name of the combined structure to use in MODIFY procedures and CHECK FILE commands.

Once you enter the COMBINE command, you can modify the combined structure.

Reference Usage Notes for the COMBINE Command

All MODIFY code compiled in releases prior to 5.2.0 must be re-compiled.

Reference Error Messages for the COMBINE Command

```
(FOC???) MAXIMUM NUMBER OF 'COMBINES' EXCEEDED. CLEAR SOME AND
RE-ENTER:
```

The number of separate COMBINE commands exceeds the current limit of 63.

Compiled DEFINES

Compiling DEFINE expressions into machine code provides faster processing. A DEFINE compiler was introduced in the UNIX and Windows NT environments in iWay5.1.0.

The MVS and VM environments have had an expression compiler for several releases. A new DEFINE compiler has been introduced on MVS, VM, and USS in iWay5.2.0.

On MVS and VM two expression compilers are available. By issuing the appropriate command you can select either one of them or disable expression compilation:

- The new DEFINE compiler compiles only those expressions that are found in DEFINE fields referenced in TABLE requests, but it provides much faster execution of those expressions than the old compiler. It provides no compilation for COMPUTE expressions in TABLE or MODIFY procedures. Compilation takes place at TABLE runtime. This compiler is invoked in the same way as the compiler on UNIX and Windows NT, by issuing the command SET DEFINES = COMPILED.
- The compiler available in prior releases is invoked with the command SET COMPUTE = NEW (the default). This compiler provides expression compilation for DEFINE, IF, WHERE, and COMPUTE commands in both TABLE and MODIFY procedures. Under this compiler, expressions are compiled at DEFINE time. Therefore, compilation may be invoked for expressions that are never actually used in a request.

Among the benefits of the new DEFINE compiler are:

- Compilation of only those expressions that are actually used in the TABLE request.
- Much faster execution of expressions containing complex calculations on long packed fields.
- Compilation of date expressions.

Once the new DEFINE compiler is invoked, any request that uses a DEFINE expression causes the expression to be compiled and then loaded into the system. For each record of the request that needs the computation, the system executes the generated code. This compiler is most effective with TABLE requests that include a large number of DEFINE fields and read a large number of records because the speed of evaluation per record in such requests offsets the extra compilation and load steps.

Syntax How to Compile DEFINE Expressions

Issue the following command in a server profile, FOCPARM, a FOCEXEC, or at the command line

```
SET DEFINES = {COMPILED|OLD}
```

where:

COMPILED

Implements expression compilation at request runtime, compiling only those DEFINES that are used in the request. COMPILED is the default value in the USS environment on the Mainframe.

OLD

In the UNIX and Windows NT environments, indicates that expression compilation will not take place. In the Mainframe environment, the OLD setting leaves expression compilation up to the control of the current SET COMPUTE value. OLD is the default value in the MVS, VM, UNIX, and Windows NT environments.

Note: Linux running in any operating environment (including the Mainframe) uses the UNIX version of the compiler and has the same default value.

Syntax How to Query Compiled DEFINE Expressions

Issue the following command to query the current setting:

```
? SET DEFINES
```

Reference Interaction Between SET DEFINES and SET COMPUTE on MVS and VM

Two expression compilers are available on MVS and VM, but only one can be activated for any request. Activating either compiler automatically deactivates the other compiler:

- Issuing the SET DEFINES=COMPILED command activates the new compiler and deactivates the older compiler.
- Issuing the SET DEFINES=OLD command deactivates the new compiler and automatically activates the old compiler.
- Issuing the SET COMPUTE command with either the OLD or NEW setting deactivates the new compiler. The OLD setting also deactivates the old compiler.

Therefore, you can select either compiler by issuing the SET DEFINES command; DEFINES=COMPILED selects the new compiler, DEFINES=OLD selects the old compiler. To turn compilation off, issue SET COMPUTE=OLD.

The new compiler is recommended for TABLE requests that include a large number of DEFINE fields (especially those that use packed arithmetic or date expressions) and read a large number of records.

The old compiler is recommended for MODIFY requests and those TABLE requests that use COMPUTE rather than DEFINE fields.

If a TABLE request uses both COMPUTE and DEFINE fields, it is almost certain to execute faster using the new compiler, especially if it retrieves a large number of records. Also, if the DEFINE fields use packed arithmetic (especially with long packed fields) or date expressions, the new compiler is likely to provide the most benefit.

Reference Usage Notes for Compiled DEFINES

- Any expression that cannot be compiled runs without compilation. This does not affect compilation of other expressions.
- The SET DEFINES command is not supported in an ON TABLE phrase.
- SET DEFINES creates a pool boundary when used in conjunction with Pooled Tables.
- If compilation is not possible because of environmental conditions, the processing will be handled without compilation. No message will be generated indicating that compilation did not take place. To determine whether compilation did take place, issue the ? COMPILE command.

Controlling Report Page Numbering

The SET FOCFIRSTPAGE command enables you to designate the first page number on a report. You can set FOCFIRSTPAGE to a specific number or the value of a Dialogue Manager variable. The &FOCNEXTPAGE variable enables you to establish consecutive page numbering across multiple reports.

When a report is processed, the variable &FOCNEXTPAGE is set to the number following the last page number in the report. This value can then be used as the first page number in a subsequent report, making the report output from multiple requests more useful and readable.

Consecutive page numbering can span multiple remote procedure calls (RPCs) and -INCLUDE commands.

If TABPAGENO is used in a request with FOCFIRSTPAGE, it correctly reflects the page number set by FOCFIRSTPAGE.

Syntax **How to Set the First Page Number for a Report**

In an RPC or a server profile

```
SET FOCFIRSTPAGE = {n|&var}
```

In a TABLE request

```
ON TABLE SET FOCFIRSTPAGE {n|&var}
```

where:

n

Is the one to six digit number to be assigned to the first page of report output. The default value is 1.

&*var*

Is a Dialogue Manager variable whose value is used as the first page number on the report. &FOCNEXTPAGE is a system variable whose value is one greater than the last page on the prior report.

Example **Setting the Number of the First Page on a Report**

This example runs two report requests, each of which uses TABPAGENO in its heading:

- The first report displays a list of movies.
- The second report displays movies with specific ratings. The SET FOCFIRSTPAGE command prior to the second report causes it to start with the next consecutive page number after the end of the first report.

Controlling Report Page Numbering

The following procedure contains both report requests:

```
TABLE FILE MOVIES
HEADING
"MOVIES BY CATEGORY AND DIRECTOR: PAGE <TABPAGENO "
" "
PRINT RATING TITLE
BY CATEGORY BY DIRECTOR
WHERE CATEGORY EQ 'ACTION' OR 'MUSICALS' OR 'COMEDY' OR 'CHILDREN'
WHERE DIRECTOR NE ' '
WHERE RATING NE 'NR'
END
-RUN
```

```
SET FOCFIRSTPAGE=&FOCNEXTPAGE
```

```
TABLE FILE MOVIES
HEADING
"MOVIES APPROPRIATE FOR CHILDREN: PAGE <TABPAGENO "
" "
PRINT TITLE
BY CATEGORY BY DIRECTOR BY RATING
WHERE CATEGORY EQ 'ACTION' OR 'MUSICALS' OR 'COMEDY' OR 'CHILDREN'
WHERE DIRECTOR NE ' '
WHERE RATING EQ 'G' OR RATING CONTAINS 'PG'
END
```

The first report has pages 1 and 2. The output is:

MOVIES BY CATEGORY AND DIRECTOR: PAGE 1

CATEGORY	DIRECTOR	RATING	TITLE
-----	-----	-----	-----
ACTION	MCDONALD P.	R	RAMBO III
	SCOTT T.	PG	TOP GUN
	SPIELBERG S.	PG	JAWS
	VERHOVEN P.	R	ROBOCOP
		R	TOTAL RECALL
CHILDREN	BARTON C.	G	SHAGGY DOG, THE
	DISNEY W.	G	BAMBI
	GEROMINI	G	ALICE IN WONDERLA
COMEDY	ABRAHAMS J.	PG	AIRPLANE
	ALLEN W.	PG	ANNIE HALL
	BROOKS J.L.	R	BROADCAST NEWS
	HALLSTROM L.	PG13	MY LIFE AS A DOG
	MARSHALL P.	PG	BIG
	ZEMECKIS R.	PG	BACK TO THE FUTUR
MUSICALS	ATTENBOROUGH R.	PG13	CHORUS LINE, A
	FOSSE B.	PG	CABARET

MOVIES BY CATEGORY AND DIRECTOR: PAGE 2

CATEGORY	DIRECTOR	RATING	TITLE
-----	-----	-----	-----
MUSICALS	FOSSE B.	R	ALL THAT JAZZ
	JEWISON N.	G	FIDDLER ON THE ROOF

The second report starts on page 3. The output is:

MOVIES APPROPRIATE FOR CHILDREN: PAGE 3

CATEGORY	DIRECTOR	RATING	TITLE
-----	-----	-----	-----
ACTION	SCOTT T.	PG	TOP GUN
	SPIELBERG S.	PG	JAWS
CHILDREN	BARTON C.	G	SHAGGY DOG, THE
	DISNEY W.	G	BAMBI
	GEROMINI	G	ALICE IN WONDERLAND
COMEDY	ABRAHAMS J.	PG	AIRPLANE
	ALLEN W.	PG	ANNIE HALL
	HALLSTROM L.	PG13	MY LIFE AS A DOG
	MARSHALL P.	PG	BIG
	ZEMECKIS R.	PG	BACK TO THE FUTURE
MUSICALS	ATTENBOROUGH R.	PG13	CHORUS LINE, A
	FOSSE B.	PG	CABARET
	JEWISON N.	G	FIDDLER ON THE ROOF

DATASET for a Segment in a Master File

In addition to having a DATASET attribute at the file level in a FOCUS Master File, you can add the DATASET attribute to a segment declaration to specify the physical file name for a LOCATION segment or a cross-referenced segment with field redefinitions. In addition, the DATASET attribute permits you to bypass the FOCUS search mechanism for default data source location. DATASET eliminates the need to allocate data sources using JCL, FILEDEF, DYNAM, and USE commands.

Note: This feature has Candidate for Release status in iWay 5.2.0.

User allocation and system specific behavior is as follows:

Platform	User allocation command
CMS/UNIX/Windows NT	FILEDEF
TSO	DYNAM ALLOC or TSO ALLOC

Note: The MODIFY FIND function is not supported with the DATASET attribute. To use FIND with a data source, you must manually allocate the data source.

DATASET Behavior in FOCUS Data Sources

The DATASET attribute can be used on the file or segment level of a FOCUS Master File. If a USE command or explicit allocation is issued for the file, a warning is issued that the DATASET attribute will be ignored.

The segment with the DATASET attribute must be either a LOCATION segment or a cross-referenced segment. For cross-referenced segments:

- If field declarations are specified for the cross-referenced fields, the DATASET attribute is the only method for specifying a physical file because FOCUS will not read the cross-referenced Master File and, therefore, will not be able to pick up its DATASET attribute if one is specified.
- If field declarations are not specified for the cross-referenced fields, it is better to place the DATASET attribute at the file level in the cross-referenced Master File as FOCUS will read and parse the cross-referenced Master File. In this case, specifying different DATASET values at the segment level in the host Master File and the file level of the cross-referenced Master File causes a conflict, resulting in a (FOC1998) message.

If DATASET is used in a Master File whose data source is managed by the FOCUS Database Server, the DATASET attribute is ignored on the server side because the FOCUS Database Server does not read Master Files for servicing table requests.

The DATASET attribute in the Master File has the lowest priority:

- A user's explicit allocation overrides DATASET attributes.
- The USE command for FOCUS data sources overrides DATASET attributes and explicit allocations.

Note: If a DATASET allocation is in effect, a CHECK FILE command must be issued in order to override it by an explicit allocation command. The CHECK FILE command will de-allocate the allocation created by DATASET.

Syntax

How to Use the DATASET Attribute

For a LOCATION segment:

```
SEGNAME=segname, SEGTYPE=segtype, PARENT=parent, LOCATION=filename1,
DATASET=' filename2 [ON sinkname]' , $
```

For a cross-referenced segment:

```
SEGNAME=segname, SEGTYPE=segtype, PARENT=parent, [CRSEGNAME=crsegname,]
[CRKEY=crkey,] CRFILE=crfile, DATASET=' filename2 [ON sinkname]',
FIELD=...
```

where:

filename2

Is the platform-dependent physical name of the data source.

sinkname

Indicates that the data source is located on the FOCUS Database Server. This attribute is valid for FOCUS data sources.

On MVS, the syntax is:

```
{DATASET|DATA}=' qualifier.qualifier ...'
```

or

```
{DATASET|DATA}=' ddname ON sinkname'
```

On CMS, the syntax is:

```
{DATASET|DATA}=' filename filetype filemode [ON sinkname]'
```

On UNIX, the syntax is:

```
{DATASET|DATA}=' path/ filename'
```

On Windows, the syntax is:

```
{DATASET|DATA}=' path\ filename'
```

Example Allocating a FOCUS Segment Using the DATASET Attribute

On MVS:

```
FILE = ...
SEGNAME=BODY, SEGTYPE=S1, PARENT=CARREC, LOCATION=BODYSEG,
DATASET= 'USER1.BODYSEG.FOCUS',
FIELDNAME=BODYTYPE, TYPE, A12, $
FIELDNAME=SEATS, SEAT, I3, $
FIELDNAME=DEALER_COST, DCOST, D7, $
FIELDNAME=RETAIL_COST, RCOST, D7, $
FIELDNAME=SALES, UNITS, I6, $
```

On UNIX/USS:

```
FILE = ...
SEGNAME=BDSEG, SEGTYPE=KU, CRSEGNAME=IDSEG, CRKEY=PRODMGR,
CRFILE=PERSFILE, DATASET= '/u2/prod/user1/idseg.foc',
FIELD=NAME, ALIAS=FNAME, FORMAT=A12, INDEX=I, $
```

On MVS with SU:

```
FILE = ...
SEGNAME=BODY, SEGTYPE=S1, PARENT=CARREC, LOCATION=BODYSEG,
DATASET= 'BODYSEG ON MYSU',
FIELDNAME=BODYTYPE, TYPE, A12, $
FIELDNAME=SEATS, SEAT, I3, $
FIELDNAME=DEALER_COST, DCOST, D7, $
FIELDNAME=RETAIL_COST, RCOST, D7, $
FIELDNAME=SALES, UNITS, I6, $
```

Reference DATASET Messages

(FOC1876) LOCATION IS NOT SPECIFIED FOR THE SEGMENT WITH DATASET:
segment-name

DATASET specification on a segment/class level requires LOCATION to associate physical data with a logical name.

FOC1876 displays if a DATASET attribute appears on a SEGMENT level with no LOCATION or cross-referenced file involved.

(FOC1877) LOCATION WITH DATASET; INCONSISTENT DATASET VALUE FOR:
location-file

Multiple segments in a structure can specify the same LOCATION name. These LOCATION attributes cannot point to multiple DATASET values. This also applies to cross-referenced segments pointing to the same cross-referenced file.

FOC1877 displays if multiple segments are stored in the same LOCATION, and DATASET specifications are in conflict.

- (FOC1998) `CONFLICTING DATASET SPECIFICATIONS FOR cross-referenced-file`
Different DATASET values are specified at the FILE level of a cross-referenced file, and at the SEGMENT level of the host file.
FOC1998 displays when the DATASET value at the FILE level of a cross-referenced file, and the DATASET value at the SEGMENT level in the host file do not match.
- (FOC1999) `INCORRECT 'ON' FILE NAME IN DATASET FOR master (ddname-in-dataset)`
When DATASET on file or segment level is used to specify sink machine as a data source (<filename> ON <sinkname> syntax), the file name must match the MFD name for file-level datasets, and LOCATION name for segment-level datasets.
FOC1999 displays if the name specified with the 'ON sink' syntax is not the Master File name for a file level DATASET, and not the LOCATION name for a segment level DATASET.

Decimal Alignment of Headings

Numbers on separate lines of a heading, footing, subheading, or subfooting can be displayed so that their decimal points align, even if they have different numbers of decimal places. The decimal alignment works regardless of the type of spacing used by the font, proportional or fixed.

Decimal alignment is supported for styled reports in the following formats:

- HTML.
- PDF.
- PostScript.

Note: This feature has Candidate for Release status in iWay 5.2.0.

Syntax **How to Specify Decimal Alignment of a Heading or Footing**

To align the decimal points, you define the width of the decimal item, then you measure how far in from the right side of a column you want to position the decimal point. This places the decimal point in the same position in a column, regardless of the number of decimal places displayed to its right.

`TYPE=hcomponent, ITEM=n, WIDTH=w, JUSTIFY=DECIMAL(m), $`

where:

hcomponent

Specifies the type of heading component that should display with decimal alignment:

TABHEADING specifies the report heading.

TABFOOTING specifies the report footing.

HEADING specifies the page heading.

FOOTING specifies the page footing.

SUBHEAD specifies a sort heading.

SUBFOOT specifies a sort footing.

n

Specifies the part of the heading or footing to align by its position within the heading or footing. Note that several other heading and footing subcomponents can also be used to select a specific item.

w

Specifies the width of the item to be aligned in the units specified by the UNITS parameter (inches by default).

m

Is the measurement expressed in the units specified by the UNITS parameter (inches by default), which specifies how far in from the right side of a column you want to place your decimal point. With this specification, you can locate the decimal point in the same position within a column, regardless of the number of decimal places displayed to its right. Text can also be aligned using this technique. The rightmost character aligns in the position immediately to the left of the decimal point.

The measurement must be a portion of the width you have specified for this item.

Example **Aligning Data and Text in a Multi-Line Heading or Footing**

In the following freeform report, content is defined entirely in the sort heading, where text and data are stacked to support comparison among regions.

The first line of each subheading displays the name of the region. The next three lines display the following numbers decimally aligned. The decimal point displays one inch from the right of a two-inch column:

- Unit sales with no decimal places.
- Dollar sales with two decimal places.
- Budgeted sales with five decimal places.

```

DEFINE FILE GGSALES
JUNITS/D15.0 = UNITS;
JDOLLARS/D15.2 = DOLLARS;
JBUDDOLLARS/D15.5 = BUDDOLLARS;
END

SET SQUEEZE = ON

TABLE FILE GGSALES
BY REGION NOPRINT SUBHEAD
"SALES FOR: <REGION "
" "
"Units: <JUNITS"
"Dollar Sales: <JDOLLARS"
"Budgeted Sales: <JBUDDOLLARS"
ON TABLE SET PAGE-NUM OFF
ON TABLE HOLD FORMAT HTML
ON TABLE SET HTMLCSS ON

ON TABLE SET STYLE *
TYPE = REPORT, GRID=OFF, $
TYPE=SUBHEAD, ITEM=1, WIDTH=1.5, $
TYPE=SUBHEAD, ITEM=2, WIDTH=2, JUSTIFY=DECIMAL(1), $
END

```

Decimal Alignment of Headings

In order for the second item to align, the first item must have a WIDTH specification so that it will take up a set amount of space. The rightmost character of the region name aligns over the digit to the left of the decimal point in the numeric values. The output is:

```
SALES FOR:                Midwest

Units:                    905,045.
Dollar Sales:             11,400,665.00
Budgeted Sales:          11,194,373.00000

SALES FOR:                Northeast

Units:                    916,675.
Dollar Sales:             11,392,310.00
Budgeted Sales:          11,576,932.00000

SALES FOR:                Southeast

Units:                    935,232.
Dollar Sales:             11,710,379.00
Budgeted Sales:          11,807,981.00000

SALES FOR:                West

Units:                    932,039.
Dollar Sales:             11,652,957.00
Budgeted Sales:          11,641,513.00000
```

Reference Usage Notes for Decimal Alignment

- To use decimal alignment in an HTML report, you must turn on internal Cascading Style Sheets (SET HTMLCSS ON).
- Decimal alignment is not supported in requests using ACROSS or OVER.

Aligning Decimals in a Multi-Line Heading or Footing

The ability to align heading content in a multi-line heading based on width and justification values has special benefit in reports that contain data with different numbers of decimal places. For example, if a figure is in dollars, it is formatted with a decimal point and two places for zeroes, if in Swiss Francs, it is formatted with a decimal place and four zeroes, if in Yen, the decimal is at the end with no zeroes. In addition, sometimes the currency doesn't vary, but the number of digits of decimal precision varies.

By aligning the decimal points in a vertical stack, you can more easily read and compare these numbers, as illustrated in the following table:

Floating decimal points		Aligned decimal points	
Bond Face	Value	Bond	Face Value -----
-----	-----	Galosh Ltd.	22375.5784596
Galosh Ltd	22375.5784596	Mukluk Inc.	1212345.457
Mukluk Inc.	1212345.457	Overshoe Inc.	232.45484
Overshoe Inc.	232.45484		

The technique uses a width specification for the item that contains decimals, combined with a variation on standard left/right/center justification to achieve the proper decimal alignment.

Procedure **How to Measure for Column Width and Decimal Alignment**

Measuring Width. Determining the width of a heading or footing item is a two-step process:

1. Identify the maximum number of characters in a text string or field.
For text strings, simply count the characters. For a field, refer to the format specification in the Master File or in a command such as DEFINE.
2. Measure the physical space in units (for example, in inches) that is required to display the number of characters identified in step 1, based on the size of the font you are using. For example, the following value of the COUNTRY field would measure as follows:

Font	Font size	Comparison	Inches
Helvetica	10	England	.5
Times New Roman	10	England	.44
Courier	10	England	.56

Tip: Consider using a consistent set of fonts in your reports to make your measurements reusable.

Measuring for Decimal Alignment. After you have determined the width of an item column, you can do a related measurement to determine the physical space required to display decimal data with a varying number of digits to the right of the decimal point.

1. Determine the maximum number of decimal places you need to accommodate to the right of the decimal place, plus the decimal point itself.
2. Measure the physical space in units (for example, in inches) that is required to display the number of characters identified in step 1, based on the size of the font you are using.

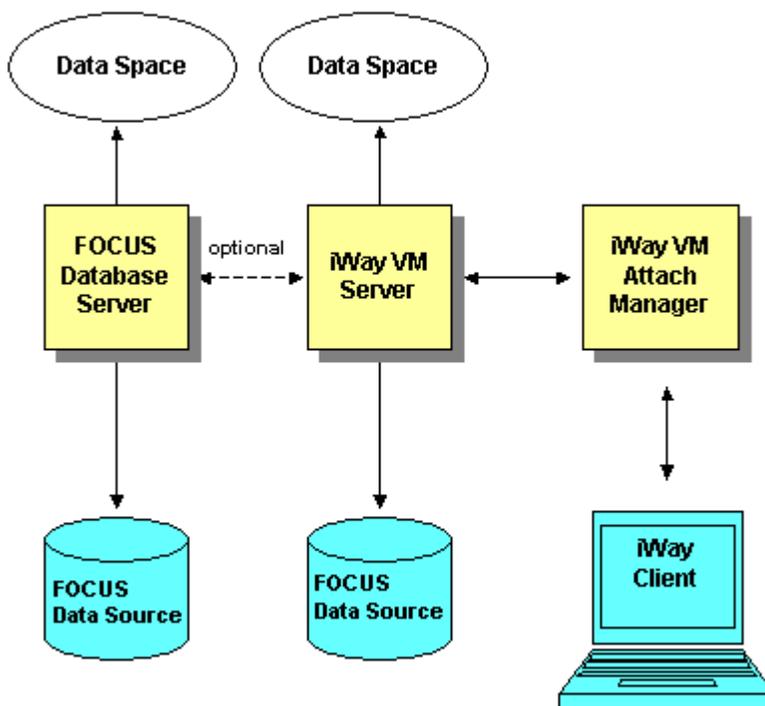
Dynamic Data Space I/O

iWay VM Servers and FOCUS Database Servers (sink machines) can now cache FOCUS data sources. This produces performance gains when running multiple requests against the same data source.

iWay Communications Overview

DYNAM DSIO requires a persistent connection while executing multiple requests against the same data source to achieve performance gains.

The following diagram illustrates connections and configuration for DYNAM DSIO when it is enabled:



Using Data Spaces

A data space is a block of virtual system memory. The following are limits associated with data spaces:

- Each data space can be a minimum of 16 (4K) pages and a maximum of 524,288 (4K) pages (2 GB) in size.
- One virtual machine can access up to 1,022 data spaces at one time.
- The total number of data spaces that can be allocated by a virtual machine is zero through 32,784.
- The total size of all data spaces allocated on a single virtual machine can be 8,192 GB.

Because data spaces reside in memory, reading from them is much faster than reading from a storage device. With dynamic data space I/O enabled, the first time any FOCUS database page is read, it is cached in a data space. Subsequent retrievals of the same data are done directly from the data space, reducing disk I/O.

Data spaces on a VM Server remain in memory as long as the client application does not disconnect from the server. On a FOCUS Database Server, the data spaces remain in memory as long as the server user ID does not log off or re-IPL CMS.

Dynamic Data Space Prerequisites

The virtual machine must operate in XC mode and have two XCONFIG statements in its CP directory entry that define the size and number of data spaces it can access.

There are several ways to establish XC mode:

- The systems staff can place the MACHINE XC statement in the directory of each user ID that requires the use of VM data spaces.
- The server administrator may also be able to set this directory option, depending on the VM security package installed on the system.
- The systems staff or a user can add the SET MACHINE XC statement to a virtual machine's PROFILE EXEC.

Note: After establishing XC mode, you must IPL CMS.

The following are sample XCONFIG statements needed to access data spaces:

```
XCONFIG ADDRSPACE MAXNUMBER 62 TOTSIZE 2048M
```

```
XCONFIG ACCESSLIST ALSIZE 62
```

Enabling Dynamic Data Space I/O

The DYNAM DSIO command controls whether FOCUS data sources are cached in data spaces and gives you access to information about these data spaces. FOCUS data sources have file type FOCUS, SUFFIX=FOC in the Master File, record format FIXED, and a logical record length of 4096. Only files with these properties are eligible to be cached when you enable DYNAM DSIO.

Syntax How to Activate Dynamic Data Space I/O

The DYNAM DSIO ENABLE command creates a list of files eligible for caching in data spaces. It does not create any data spaces or cache any data. These functions occur when data is actually retrieved in response to a request.

- On a VM server, the DYNAM DSIO ENABLE command can be issued from the EDASPROF EDAPROF global profile or in a remote procedure call (RPC).
- On a FOCUS Database Server, the DSIO ENABLE command can be issued from the IUCVMAIN CONFIG file or the server.s command line. (**Note:** Dynamic data space I/O is not supported for VMCF FOCUS Database Servers.)

The DSIO command can also be issued from the command line when the FOCUS Database Server is in NODISC mode.

Issue the following command to enable the specified data sources to use data spaces:

```
[DYNAM] DSIO [ENABLE] filespec
```

where:

DYNAM

Must be omitted when issuing the command in a FOCUS Database Server's IUCVMAIN CONFIG file. It must be included in the syntax in all other contexts.

ENABLE

Must be omitted when issuing the command in a FOCUS Database Server's IUCVMAIN CONFIG file. It must be included in the syntax in all other contexts.

filespec

Specifies which FOCUS data sources in the user's search list are affected. Can be one of the following:

*

Specifies all files with file type FOCUS. These files are accessed and placed on the list in the normal VM search order.

`fn`

Specifies the first file named `fn` with file type FOCUS found in the VM search order. Note: If you have a FOCUS data source with a one-letter name, DYNAM DSIO may mistake the name for a minidisk letter. In this case you can enable it using its complete file ID (filename filetype filemode). These files are accessed and placed on the list in the order specified.

`fm`

Specifies the one-character minidisk letter portion of a file mode. This specifies all files with file type FOCUS on that minidisk. Note: If you have a FOCUS data source with a one-letter name, you can enable it using its complete file ID (filename filetype filemode). These files are accessed and placed on the list in the order specified.

`fn FOCUS fm`

Specifies the file named `fn` with file type FOCUS and file mode `fm`. These files are accessed and placed on the list in the order specified.

Note: File mode 6 files will only be cached into a data space if they are on an accessed READ WRITE (R/W) minidisk. Any records retrieved prior to issuing the DYNAM DSIO ENABLE command will not be cached in the data space.

Example Enabling Dynamic Data Space I/O

The following command enables dynamic data space I/O for the FOCUS data source named CAR with file type FOCUS and file mode F:

```
DYNAM DSIO ENABLE CAR FOCUS F
```

The following command enables dynamic data space I/O for the first FOCUS data source named CAR with file type FOCUS in the standard CMS search list:

```
DYNAM DSIO ENABLE CAR
```

Example Enabling Dynamic Data Space I/O on a FOCUS Database Server

This sample contains the portion of the IUCVMAIN CONFIG file that refers to the DSIO feature.

```

*
* Server will cache FOCUS databases in a DATASPACE. Do not separate
* minidisks by blanks. Define minidisks separated by commas.
* Multiple statements must be specified for multiple specific FOCUS
* files. If only minidisks are specified, these can be all specified in
* one DSIO entry. You may mix minidisk entries and specific file entries
* For instance, you might cache CAR FOCUS A, EMPLOYEE FOCUS B, and all
* FOCUS databases on C,D,E. This would be 3 DSIO entries:
* 1 for CAR FOCUS A, 1 for EMPLOYEE FOCUS B, and 1 for C,D,E.
*DSIO *                Indicates to CACHE all FOCUS databases on all
*                    minidisks.
*DSIO                With no options is the same as all FOCUS databases
*                    on all minidisks.
*DSIO B,C,D,E,F      Cache only FOCUS databases on minidisks specified
*DSIO EMPLOYEE       Enables cache for the first EMPLOYEE FOCUS * file
*                    found in A-Z search order. Same as EMPLOYEE FOCUS
*DSIO CAR FOCUS      Same as CAR FOCUS *
*DSIO CAR FOCUS T    Enables cache for ONLY the CAR FOCUS T file.
*                    If filetype is specified it must be FOCUS.
*                    Filetype MUST be specified to specify filemode.

```

Syntax How to Query Dynamic Data Spaces

The following command displays a list of active data spaces for the user ID:

```
DYNAM DSIO QUERY
```

The next command returns statistical information about the amount of data space used:

```
DYNAM DSIO LIST
```

The following command displays DYNAM DSIO command help information:

```
DYNAM DSIO ?
```

Example Querying Active Data Spaces

The following example shows sample output of the DYNAM DSIO QUERY command:

ASIT	ALET	4K_PAGES	DSLISIT	SPACE_NAME
26609F4000000203	32768	00000000	USER1:BASE
300A5E40000000A6	1024	00A83018	USER1:CAR-FOCUS-A_RW
040C720000000153	01000002	281232	00000000	SFSDS:0000001117DSU
1B6FDA8000000128	01000003	281232	00000000	SFSDS:0000001115DSU
1B6FD1C000000083	01000004	281232	00000000	SFSDS:0000001116DSU

The following information is displayed:

ASIT

Address Space Identification Token. CP uses this token to indicate the location where the data space starts in central storage.

ALET

Access List Entry Token. CMS uses this token to indicate the location where the data space starts as accessed by CMS.

4K_PAGES

Amount of storage used. This value indicates the size of the data space in 4K pages.

DSLISIT

Data Space List. FOCUS uses this token to indicate the location where the data source starts in the data space.

Note: This token is only populated with a value for FOCUS data sources that have had data loaded into the data space.

SPACE_NAME

Data space identification. This value identifies the data source and its access mode in the data space. For example:

USER1 : CAR-FOCUS-A_RW

USER1 is the USERID of the data space.

CAR-FOCUS-A is the file name, file type, and file mode of the FOCUS data source.

_RW indicates that the file is in R/W mode.

Example Listing Data Space Usage Statistics

The following example shows the output of the DYNAM DSIO LIST query:

FILENAME	FILETYPE	FM	*	MDISKRD	MDISKWT	SPACERD	SPACEWT	SPACEUD	DISKROF	DISKWOF
CAR	FOCUS	A1	W	8	3	9	8	11	0	0

The following information is displayed:

FILENAME

The CMS file name of the data source.

FILETYPE

The CMS file type of the data source.

FM

The CMS file mode of the data source.

*

The access mode of the data source, W for Write or R for Read.

MDISKRD

Minidisk reads counter.

This counter indicates the number of records read from the data source.

MDISKWT

Minidisk writes counter.

This counter indicates the number of records written to the data source.

SPACERD

Data space reads counter.

This counter indicates the number of records read from the data space.

Note: A high number in the counter is a good indication that the data space has been heavily utilized.

SPACEWT

Data space writes counter.

This counter indicates the number of records written to the data space.

Note: The counter indicates the number of unique records written to the data space.

SPACEUP

Data space update counter.

This counter indicates the number of records updated in the data space.

Note: The counter indicates the number of unique records updated in the data space.

DISKROF

Disk Read Overflow counter.

This counter indicates the number of records that could not be written to the data space on a Read operation. An overflow indicates the data space is full.

Note: In order to read records from a data space, the records must first be written/loaded into the data space.

DISKWOF

Disk Write Overflow counter.

This counter indicates the number of records that could not be written to the data space, after it became full.

Example **Displaying DYNAM DSIO Command Help Information**

The following example shows the output of the DYNAM DSIO ? query:

```
THE FOLLOWING OPTIONS ARE AVAILABLE
ENABLE  <* or A,B,C,...,Z> or FN <FOCUS FM>
LIST    - display dataspace counters
QUERY   - enhanced CP QUERY SPACES and CP DISPLAY ACCLIST
?       - displays this information
```

Dynamic Data Space I/O Startup Processing

Before caching files into data spaces, DYNAM DSIO needs information about the size and number of data spaces the user is allowed. It can find this information by testing the system, which can require a significant amount of time and system resources. To save this startup time, you can supply this information to DYNAM DSIO.

Therefore, DYNAM DSIO starts up as follows:

1. It looks for a file named DSQUERY ERRORS in the user Ides search order. This file can contain parameters that specify the number and size of the data spaces the user ID can allocate. This information should match the information found in the XCONFIG directory statements for that user ID. If the information does not match, the XCONFIG statements in the user.s CP directory take precedence.

Note: A sample DSQUERY ERRORS file is delivered on the iWay tape, but it has these parameters commented out because they are site and user ID specific.

2. If the DSQUERY ERRORS file is found and contains the necessary parameters, DYNAM DSIO uses the information found in the file. If the information does not match the limits established by the XCONFIG directory statements, DYNAM DSIO will allocate 125% of the size of the FOCUS data source if this amount does not exceed the amount specified in the user ID.s CP directory.

The steps DYNAM DSIO takes to allocate a data space in this case are:

- a. Check the CP directory for the number and size of allowed data spaces.
- b. Check the size of the FOCUS data source. The data space will be allocated based upon the size of the FOCUS data source and the allowable data space size in the CP directory.

For example, if the FOCUS data source is 100M in size and the allowable data space size is 500M, then the data space will be created with a size of 125M.

If, on the other hand, the allowable data space size is 100M and the FOCUS data source is 90M, the data space will be created with a size of 100M. DYNAM DSIO cannot create a data space that is 125% of the size of the data source because that would be larger than the allowable data space size. It creates the largest data space possible based on the allowable size.

As these examples illustrate, DYNAM DSIO can never allocate more space than the allowable data space size specified in the CP directory.

Allocating a smaller data space may not give you the optimal performance, but it should not interrupt your session.

3. If the file is not found or does not specify the needed parameters, DYNAM DSIO tests the system to see what the actual limits are before it allocates any data space.

Example Sample DSQUERY ERRORS File

* The following lines are commented out. To enable remove the asterisk in
* column 1. These entries should match your XCONFIG entries in the VM
* USER directory.

*
* If no values are present, these will be obtained via a QUERY PROCESS
* which may take a bit of real time, (seconds to several minutes).

* 0-32767 1-8192G
* XCONFIG ADDRSPACE MAXNUMBER 50 TOTSIZE 1024M

* XCONFIG ACCESSLIST ALSIZE 62
* 1-1022 defaults to 62

*MAXNUMBER 50
*TOTSIZE 1024M
*ALSIZE 62
*

* The following values shown are the default values, if none are coded.

*
* The following line defines the FOCUSORT FOCUSTEMP value.
* It may be defined as M for megabytes, G for gigabytes, % (percent) as
* a percent of the total size, as given in the above XCONFIG TOTSIZE
* directory entry, or pages, with no qualifier present.
*

FOCUSTEMP 5%

*
* The following line defines the percentage allocated for a writable
* FOCUS database. It must be defined as a % (percent) and should be
* larger than 100 percent. This percent is used to define a dataspace
* for R/W FOCUS databases.
*

WRITESIZE 125%

*
* The following line defines the minimum size of a dataspace. Small
* FOCUS databases will use this size for dataspace allocation.
* It may be defined as M for megabytes, G for gigabytes, % (percent) as
* a percent of the total size, as given in the above XCONFIG TOTSIZE
* directory entry, or pages, with no qualifier present.
*

MINSIZE 4M

DYNAM DSIO Usage

- When the DYNAM DSIO ENABLE command is issued, all of the FOCUS data sources specified in the command are enabled and included in the access list. However, when multiple FOCUS data sources with the same name on separate minidisks are enabled, FOCUS determines which data source is actually used.
- The order of data sources in the DYNAM DSIO list depends on the order in which DYNAM DSIO ENABLE commands were issued and the parameters used to select data sources. However, the order of the list does not affect normal FOCUS behavior. FOCUS accesses the data source with file mode A unless a USE command is issued to specify a different file mode.

In the following example, the DYNAM DSIO LIST would list the CAR data source first and the EMPLOYEE data source second even though the normal VM search order would have defaulted to the opposite order.

```
DYNAM DSIO ENABLE CAR FOCUS D
```

```
DYNAM DSIO ENABLE EMPLOYEE FOCUS C
```

- If you create a new FOCUS data source using HOLD FORMAT FOCUS, it will not be in the list of enabled files unless you reissue the DYNAM DSIO ENABLE command to enable it.

DYNAM DSIO Processing

The following illustrates a typical use of the DYNAM DSIO ENABLE feature.

1. First, issue the DYNAM DSIO ENABLE command:

```
DYNAM DSIO ENABLE CAR
```

2. To see a list of what is enabled, issue the DYNAM DSIO QUERY command:

```
ASIT          ALET          4K_PAGES  DSLIST    SPACE_NAME
26609F400000203 .....      32768  00000000  USER1:BASE
300A5E4000000A6 .....      1024  00000000  USER1:CAR-FOCUS-A_RW
040C72000000153 01000002  281232  00000000  SFSDS:0000001117DSU
1B6FDA8000000128 01000003  281232  00000000  SFSDS:0000001115DSU
1B6FD1C000000083 01000004  281232  00000000  SFSDS:0000001116DSU
```

3. Next, load data into the data space by issuing a retrieval command against an enabled FOCUS data source:

```
TABLE FILE CAR
PRINT ...
END
```

4. Then, to see the usage, issue the DYNAM DSIO LIST command:

```
FILENAME FILETYPE   FM * MDISKRD MDISKWT SPACERD SPACEWT SPACEUD DISKROF DISKWOF
CAR        FOCUS     A1 W      8        3        9        8        11       0        0
```

5. Now that CAR FOCUS has data loaded into the data space, its DSLIST token is now populated in the DYNAM DSIO QUERY:

ASIT	ALET	4K_PAGES	DSLIST	SPACE_NAME
26609F4000000203	32768	00000000	USER1:BASE
300A5E40000000A6	1024	00A83018	USER1:CAR-FOCUS-A_RW
040C720000000153	01000002	281232	00000000	SFSDS:0000001117DSU
1B6FDA8000000128	01000003	281232	00000000	SFSDS:0000001115DSU
1B6FD1C000000083	01000004	281232	00000000	SFSDS:0000001116DSU

Reference **DYNAM DSIO Error Messages**

All DYNAM DSIO messages are WARNING messages:

- (FOC32158) UNABLE TO PARSE PARMS FOR DSIO AT LINE 105 IN IUCVMMAIN CONFIG
The parms or options for the keyword specified at the line specified were found to be in error. Check the examples in the CONFIG file. You may have specified an invalid word or extra parameter.
- (FOC32160) WARNING: DATABASE ATTRIBUTES DO NOT MATCH CURRENT VALUES
Database attributes are stored in the data space. These attributes do not match the current attributes obtained. The data space will be destroyed and recreated based on the new attributes.
- (FOC32161) WARNING: DATASPACE OVERFLOW &1 &2 &3
The size of the data space was insufficient to contain the database record requested. Processing continues, but the database record was not cached. You may want to increase the WRITESIZE parameter in DSQUERY ERRORS.
Note(s) for error message (FOC32161)
This error is displayed, on the sink machine console, the first time that FOCUS detects a read overflow to an accessed data space. It is also displayed once on the first time that a write overflow is detected to an accessed data space.
- (FOC32162) QUERY ERROR. USERID MAY NOT BE CONFIGURED FOR DATASPACE USE.
This userid may not be authorized for data space usage. See your directory administrator for data space authorization and add a personal copy of DSQUERY ERRORS to your "A" disk for faster data space access.

(FOC32163) DATASPACE USE REQUIRES USERID TO BE IN XC MODE

DATASPACE usage requires the virtual machine to be in XC mode. Issue 'SET MACHINE XC' from the CMS command line and RE-IPL CMS.

(FOC32164) FOCUS FILE NOT FOUND ON ANY ACCESSED DISK

A command was entered which specified a FOCUS database, but the database was not found.

(FOC32165) NO DATASPACE LIST EXISTS

A command was entered which examines the DATASPACE list, but no list is currently defined.

Exit On Error

Using the ERROROUT command, you can control how an iWay server responds to error conditions encountered in a procedure.

The ERROROUT setting is ignored in an interactive session. If you issue the ERROROUT command in an interactive setting, the following message is returned:

```
SETTING IGNORED IN INTERACTIVE MODE
```

Controlling Batch Error Processing

When ERROROUT is set to ON, when an error message is generated the procedure terminates and the error message is returned to the client. In addition, the following message displays to inform the user why the program terminated:

```
Exiting due to Exit on Error
```

Warning messages do not invoke this behavior. When a warning is generated, processing continues as normal.

Users can check a Dialogue Manager variable such as &FOCERRNUM and issue the following command to terminate FOCUS and set n as the return code:

```
-QUIT FOCUS n
```

Syntax **How to Control Batch Error Processing**

`SET ERROROUT = {ON|OFF}`

where:

ON

Terminates the procedure and returns the error message to the client in the event of an error on an iWay server. This setting is ignored in an interactive session.

OFF

Does not set a return code or automatically terminate a job step or procedure in response to any error message. This is the default value.

Reference **Usage Notes for Exit on Error**

- The SET ERROROUT command can be issued in any supported profile, in the batch input stream, or in a FOCEXEC. It is not supported in a TABLE request, using ON TABLE SET syntax.
- If the user code handles return codes by using a technique such as checking &FOCERRNUM having this feature enabled will cause the program to exit and, therefore, ignore the error handling logic. In this situation, the user should set ERROROUT to OFF in order to have the error handling logic respected.

Extended Currency Symbol Support

By default, the M and N display options print the currency symbol associated with the code page being used by your system. For example, when using the U. S. code page, the M and N options display the dollar sign (\$). Extended currency symbol format options allow you to display the following currency symbols regardless of the code page used: U. S. dollar, euro, British pound, and Japanese yen. The M and N options continue to operate as in prior releases.

Extended Currency Symbol Format Options

The extended currency symbol format options consist of two characters: an exclamation point followed by one of the supported upper or lower case letters. An upper case letter displays a floating currency symbol on each detail line. A lower case letter displays a fixed currency symbol to the left of the field on the first detail line of each report page. These options are valid for numeric formats (I, D, F, and P).

Syntax How To Display an Extended Currency Symbol

Use the following character combinations as the final two characters in any numeric display format:

Display Option	Description	Example
!d	Fixed dollar sign	D12.2!d
!D	Floating dollar sign	D12.2!D
!e	Fixed euro symbol	F10.2!e
!E	Floating euro symbol	F10.2!E
!!	Fixed British pound sign	D12.1!!
!L	Floating British pound sign	D12.1!L
!y	Fixed Japanese yen symbol	I9!y
!Y	Floating Japanese yen symbol	I9!Y

Reference Usage Notes for Extended Currency Symbol Support

- Format specifications are limited to a total of eight characters.
- The extended currency option must be the last option in the format.
- The extended currency option cannot be used in the same format specification as M or N.
- In order to display or print the extended currency symbols, you must be sure they are supported by the fonts accessible to your emulation software or printer.
- The symbol displayed by the M and N options is dependent on the code page being used. The !d and !D options always display a dollar sign, regardless of the code page.
- Using a fixed currency symbol places the symbol only on the first line of each report page. If you use field-based reformatting to display multiple currency symbols in one report column, only the symbol associated with the first row appears. In this case, you should use floating currency symbols.
- Lower case letters are transmitted as upper case letters by the terminal I/O procedures. Therefore, the fixed extended currency symbols can only be specified in a REQUEST.

Syntax Using Extended Currency Symbols with TSO

When using the Time Sharing Option (TSO), reports with HotScreen set to OFF do not display the extended currency symbols by default. This is because the terminal input/output (I/O) procedures translate all terminal output to characters that appear in USA EBCDIC keyboard layouts and code charts.

You can change this default behavior with the following command.

```
SET TRANTERM = {ON|OFF}
```

where:

ON

Does not display extended currency symbols. ON is the default.

OFF

Displays extended currency symbols.

Example **Displaying Extended Currency Symbols**

The following request uses field-based reformatting to display the Japanese yen on the report row that represents Japan, the British pound on the row that represents England, and the euro on the row that represents Italy. Note that the comma inclusion display option (C) in the format for England is specified prior to the currency option:

```
DEFINE FILE CAR
  CFORMAT/A8 = DECODE COUNTRY('ENGLAND' 'F12.1C!L' 'JAPAN' 'D12!Y'
    ELSE 'D12.2!E');
END
TABLE FILE CAR
  PRINT SALES/CFORMAT DEALER_COST/CFORMAT
  BY COUNTRY
  WHERE COUNTRY EQ 'ENGLAND' OR 'JAPAN' OR 'ITALY'
  WHERE SALES GT 0
END
```

The output is:

COUNTRY	SALES	DEALER_COST
-----	-----	-----
ENGLAND	£12,000.0	£11,194.0
ITALY	_30,200.00	_16,235.00
JAPAN	¥78,030	¥5,512

FIELDTYPE=R

The field-level Master File attribute FIELDTYPE=R identifies a field as read-only. The update languages MODIFY and Maintain will not populate columns with this designation. If an update command is issued for a read-only field, it will be ignored by MODIFY and Maintain.

This attribute provides support for relational data sources with auto-increment columns such as IDENTIFY or timestamp columns in DB2. These columns are automatically incremented by the RDBMS.

CREATE SYNONYM will add the FIELDTYPE=R attribute for READ-ONLY fields in an RDBMS table.

Note:

- This feature has Candidate for Release status in iWay 5.2.0.
- This attribute is not supported for DB2 on VM.

FIELDTYPE=R

Syntax How to Specify a Field as Read-Only in a Master File

```
FIELD= ..., FIELDTYPE = R,$  
FIELD= ..., FIELDTYPE = RI,$  
FIELD= ..., FIELDTYPE = IR,$
```

where:

R

Indicates that the field is read-only. Any number of fields can have this attribute.

RI or IR

Indicates that the field is read-only and indexed (applies to FOCUS, VSAM, and XFOCUS data sources only).

Example Using FIELDTYPE=R

Autoinc1 is an MS-SQL table with an IDENTITY field called Control as the first field, which is a key.

The following CREATE SYNONYM command creates a Master File with the FIELDTYPE=R attribute for the Control field:

```
CREATE SYNONYM MNTAUTO FOR AUTOINC1 DBMS SQLMSS AT MSXYZ  
END
```

The generated Master File is:

```
FILE=MNTAUTO , SUFFIX=SQLMSS , $  
SEGNAME=MNTAUTO , SEGTYPE=S0 , $  
FIELD=CONTROL , CONTROL , I11 , I4 , MISSING=OFF, FIELDTYPE=R, $  
FIELD=LASTNAME , LASTNAME , A12 , A12 , MISSING=OFF, $  
FIELD=FIRSTNAME , FIRSTNAME , A12 , A12 , MISSING=OFF, $  
FIELD=ITEM , ITEM , A20 , A20 , MISSING=OFF, $  
FIELD=AMOUNT , AMOUNT , P19 , P10 , MISSING=OFF, $
```

The generated Access File is:

```
SEGNAME=MNTAUTO ,  
TABLENAME=AUTOINC1 ,  
CONNECTION=MSSXYZ ,  
KEYS=1  
, $
```

The following Maintain procedure that adds new data does not assign a value for auto-increment column called Control or reference it in the INCLUDE command:

```
MAINTAIN FILE MNTAUTO
INFER CONTROL LASTNAME FIRSTNAME ITEM AMOUNT INTO INSTK
PERFORM UPDAT1

CASE UPDAT1
COMPUTE INSTK(1).LASTNAME='PIG';
COMPUTE INSTK(1).FIRSTNAME='PORKY';
COMPUTE INSTK(1).ITEM='PORKBELLIES';
COMPUTE INSTK(1).AMOUNT=2000.00;
FOR 1 INCLUDE LASTNAME FIRSTNAME ITEM AMOUNT FROM INSTK
ENDCASE

END
```

If the record is successfully included, the RDBMS will have automatically generated a new value for Control to correspond to this new record.

If the Maintain code provides a value for Control as in the following commands, the updated value for that field is ignored:

```
COMPUTE INSTK(1).CONTROL=1234;
FOR 1 INCLUDE CONTROL LASTNAME FIRSTNAME ITEM AMOUNT FROM INSTK
```

It is up to the developer to query the data source to find out what value was generated by the RDBMS for the auto-increment field.

File Extensions for Maintain Procedures

Maintain procedures can have file extensions different from those of other FOCEXECs. For both MVS and CMS the extension is MAINTAIN. For UNIX and Windows NT it is .mnt.

Syntax How to Work With Maintain File Extensions

In order to work with Maintain procedures that have their own extensions, the MNTCON command must be issued as follows:

To execute a Maintain procedure:

```
MNTCON EX maintproc
```

To compile a Maintain procedure:

```
MNTCON COMPILE maintproc
```

This creates a compiled procedure with the filetype .fcm on UNIX and Windows NT or FOCCOMP on MVS and CMS (ddname FOCCOMP should be allocated for MVS).

To run a compiled Maintain procedure:

```
MNTCON RUN maintproc
```

where:

```
maintproc
```

Is a compiled Maintain procedure (.fcm or FOCCOMP)

Reference Usage Notes for Maintain File Extensions

- If Maintain does not find the procedure under file type MAINTAIN on VM, ddname MAINTAIN on MVS, or extension .mnt on UNIX and Windows NT, it looks for file type FOCEXEC on VM, ddname FOCEXEC on MVS, or extension .fex on UNIX and Windows NT.
- For UNIX and Windows NT, front-end tools generate .mnt as the extension for the Maintain procedure. For Mainframe FOCUS, you must save your Maintain procedure in ddname MAINTAIN on MVS, or as FILETYPE MAINTAIN on VM. The ddname MAINTAIN must be allocated for MVS iWay servers.

Example How to Define File Extensions for Maintain**In MVS JCL:**

```
//MAINTAIN DD=myqual.MAINTAIN.DATA.DISP=SHR
```

In VM:

```
test1 MAINTAIN A1
```

On Windows NT:

```
C:\mydata\test1.mnt
```

FML Hierarchies

Hierarchical relationships between fields can be defined in a Master File and automatically displayed using the Financial Modeling Language (FML). The parent and child fields must share data values and their relationship should be hierarchical. The formats of the parent and child fields must both be numeric or both alphanumeric.

For example, suppose that:

- An employee data source contains both the employee's ID and the ID of the employee's manager.

or

- A general ledger data source contains both an account number field and an account parent field.

By examining these fields, it is possible to construct the entire organization chart or chart of accounts. However, to print the chart in a traditional FML report, you must list the employee IDs or account numbers in the request syntax in the order in which they should appear on the report. If an employee or account is added, removed, or transferred, you have to change the report request to reflect this change in organizational structure. For example:

```
TABLE FILE EMPLOYEE
PRINT DEPARTMENT CURR_JOBCODE
FOR EMP_ID
999999999 OVER
222222222 OVER
.
.
.
```

With FML hierarchies, you can define the hierarchical relationship between two fields in the Master File and load this information into memory. The FML request can then dynamically construct the rows that represent this relationship and display them in the report starting at any point in the hierarchy.

Requirements for FML Hierarchies

1. In the Master File, use the PROPERTY=PARENT_OF and REFERENCE=*hierarchyfld* attributes to define the hierarchical relationship between two fields.
2. The hierarchy must be loaded into memory. If the hierarchy is defined in the Master File referenced by the FML request, the hierarchy is loaded automatically. If you want to use a hierarchy defined in a Master File that will not be either referenced in the FML request or joined to the Master File referenced in the FML request, you must issue the LOAD CHART command before issuing the FML request.

The number of charts that can be loaded is limited by available memory. Charts are automatically unloaded when FOCUS terminates.

3. In the FOR phrase of the FML request, use the GET CHILDREN or ADD phrase to retrieve the hierarchical data starting at a specific point in the hierarchy.

To use an FML hierarchy to report from a dynamic join or a Master File cross-reference:

- The join or cross-reference must be unique.
- The hierarchy file must be the cross-referenced file and the transaction file must be the host file.

As with any FML request, a tagged row is displayed even if no data is found in the file for the tag values, with a period (.) representing the missing data. You can override this convention by adding the phrase WHEN EXISTS to the definition of a tagged row. This makes displaying a row dependent upon the existence of data for the tag.

Syntax How to Specify a Hierarchy Between Fields in a Master File

FIELD=*parentfield*, ..., PROPERTY=PARENT_OF, REFERENCE=[*seg.*] *hierarchyfld*, \$

where:

seg

Is the segment location of the hierarchy field. Required if more than one segment has a field named *hierarchyfield*.

parentfield

Is the parent field in the hierarchy.

PARENT_OF

Identifies this field as the parent of the referenced field in a hierarchy. These attributes can be specified on every field. Therefore, multiple hierarchies can be defined in one Master File. However, an individual field can have only one parent. If multiple fields have PARENT_OF attributes for the same hierarchy field, the first parent found by traversing the structure in top-down, left-to-right order is used as the parent.

hierarchyfld

Is the child field in the hierarchy.

Syntax How to Assign Descriptive Captions for Hierarchy Field Values

The following attributes specify a caption for a hierarchy field in a Master File

FIELD= *captionfield*, ...,
PROPERTY=CAPTION, REFERENCE=[*seg.*] *hierarchyfld*, \$

where:

captionfield

Is the name of the field that contains the descriptive text for the hierarchy field. For example, if the employee ID is the hierarchy field, the last name may be the descriptive text to be displayed on the report in place of the ID.

A caption can be specified for every field, but an individual field can have only one caption. If multiple fields have CAPTION attributes for the same hierarchy field, the first parent found by traversing the structure in top-down, left-to-right order will be used as the caption.

PROPERTY=CAPTION

Signifies that this field contains a descriptive caption to be displayed in place of the hierarchy field values.

seg

Is the segment location of the caption field. Required if more than one segment has a field named *captionfield*.

`hierarchyfld`

Is the hierarchy field.

CAPTION is also allowed on a virtual field in the Master File:

```
DEFINE name/format=expression; , PROPERTY=CAPTION, REFERENCE=hierarchyfld , $
```

Example Defining a Hierarchy in a Master File

The CENTGL Master File contains a chart of accounts hierarchy. The field GL_ACCOUNT_PARENT is the parent field in the hierarchy. The field GL_ACCOUNT is the hierarchy field. The field GL_ACCOUNT_CAPTION can be used as the descriptive caption for the hierarchy field:

```
FILE=CENTGL , SUFFIX=FOC
SEGNAME=ACCOUNTS , SEGTYPE=S01
FIELDNAME =GL_ACCOUNT, ALIAS=GLACCT , FORMAT=A7,
TITLE=Account, FIELDTYPE=I , $
FIELDNAME =GL_ACCOUNT_PARENT, ALIAS=GLPAR , FORMAT=A7,
TITLE=Parent,
PROPERTY=PARENT_OF, REFERENCE=GL_ACCOUNT , $
FIELDNAME =GL_ACCOUNT_TYPE, ALIAS=GLTYPE , FORMAT=A1,
TITLE=Type , $
FIELDNAME =GL_ROLLUP_OP, ALIAS=GLROLL , FORMAT=A1,
TITLE=Op , $
FIELDNAME =GL_ACCOUNT_LEVEL, ALIAS=GLLEVEL , FORMAT=I3,
TITLE=Lev , $
FIELDNAME =GL_ACCOUNT_CAPTION, ALIAS=GLCAP , FORMAT=A30,
TITLE=Caption,
PROPERTY=CAPTION, REFERENCE=GL_ACCOUNT , $
```

The CENTSTMT data source contains the financial data for the account hierarchy in CENTGL. The following Master File for CENTSTMT has a unique cross-reference to the CENTGL Master File. CENTSTMT (the transaction file) is the host file. CENTGL (the hierarchy file) is the cross-referenced file:

```
FILE=CENTSTMT, SUFFIX=FOC,$
SEGNAME=SEG01, SEGTYPE=S2, $
FIELD=PERIOD, ALIAS=MONTH, FORMAT=YYM, $
FIELD=GL_ACCOUNT, ALIAS=ACCT, FORMAT=A7, INDEX=I,
TITLE='Account', $
FIELD=ACTUAL_AMT, ALIAS=AA, FORMAT=D10.0, MISSING=ON,
TITLE='Actual', $
FIELD=BUDGET_AMT, ALIAS=BA, FORMAT=D10.0, MISSING=ON,
TITLE='Budget', $
FIELD=ACTUAL_YTD, ALIAS=AYTD, FORMAT=D12.0, MISSING=ON,
TITLE='YTD,Actual', $
FIELD=BUDGET_YTD, ALIAS=BYTD, FORMAT=D12.0, MISSING=ON,
TITLE='YTD,Budget', $
SEGNAME=ACCOUNTS, SEGTYPE=DKU, PARENT=SEG01,
CRFILE=CENTGL, CRKEY=GL_ACCOUNT,$
```

Syntax

How to Load a Hierarchy From One Master File for Use With a Separate Master File

You can manually load the hierarchy data if you need to use a hierarchy defined in one Master File against a data source that is not joined to the hierarchy Master File (but that contains the same hierarchy field).

The number of charts that can be loaded is limited by available memory. Charts are automatically unloaded when FOCUS terminates.

```
LOAD CHART chartfile[.sega].hierarchyfld [FOR requestfile[.sega].fieldb]]
```

where:

`chartfile`

Is the name of the Master File that contains the hierarchy information.

`sega`

Is the name of the segment that contains the hierarchy field. The segment name is only required if a field in another segment in the structure has the same field name as the hierarchy field.

`hierarchyfld`

Is the hierarchy field. The hierarchy field is required because a Master File can define multiple hierarchies.

FOR

Loads a hierarchy defined in a Master File that will not be used in the FML report request. For example, Master File B contains the hierarchy information but Master File A will be used in the request (without a join between Master Files A and B), you would issue the following LOAD CHART command prior to the FML request:

```
LOAD CHART B.FLDB FOR A.FLDA  
TABLE FILE A ...
```

requestfile

Is the name of the Master File used in the FML request.

segb

Is the name of the segment that contains the hierarchy field values in the Master File used in the FML request. Not required if it has the same name as *sega*.

fieldb

Is the field in the Master File specified in the FML request that contains the values of the hierarchy field. Not required if it has the same name as the hierarchy field.

Note:

- If you issue the LOAD CHART command multiple times for the same hierarchy, the new hierarchy overlays the previous version in memory.
- If you issue the LOAD CHART command for a data source that is dynamically joined to the hierarchy file, you must issue the JOIN command prior to issuing the LOAD CHART command.

Syntax

How to Report From an FML Hierarchy

```
TABLE FILE filename  
{PRINT|SUM} ....  
FOR hierarchyfld  
parentvalue GET CHILD[REN] [n|ALL] [AS CAPTION|'text'] [LABEL label]  
.  
.  
.  
END
```

where:

filename

Is the name of the file to be used in the FML request. (If the hierarchy for this request cannot be loaded automatically, you must have previously issued the LOAD CHART command to load the hierarchy.)

hierarchyfld

Is the hierarchy field name. If the request references a joined structure, the name must be the field name from the host file. The alias name is not supported.

parentvalue

Is the parent value for which the children are to be retrieved.

n|ALL

Is a positive integer from 1 to 99 specifying the number of levels of the hierarchy to retrieve. If *n* is omitted, direct children are retrieved. GET CHILDREN 2 retrieves direct children and grandchildren. GET CHILDREN 99 retrieves children to 99 levels. ALL is a synonym for 99. Each child instance is printed over the next child instance. (This corresponds to the FML syntax CHILD1 OVER CHILD2 OVER ...) Successive levels of the hierarchy field are indented two spaces from the previous level.

CAPTION

Indicates that the caption values to display should be taken from the field defined as the CAPTION in the Master File.

Note: The AS CAPTION phrase is supported for tagged rows that do not use the GET CHILDREN or ADD syntax. However, the hierarchy must be defined (by specifying the PARENT_OF attribute) in order to load and display the caption values. If the hierarchy is not defined, the AS CAPTION phrase is ignored.

'text'

Is a text string to use as the caption for the hierarchy field values. The CAPTION field defined in the Master File will not be used as the caption on the report output.

label

Is an explicit row label. Each generated row is labeled with the specified label text.

Note: The hierarchy is displayed sorted by the parent field and, within parent, sorted by the hierarchy field.

Syntax How to Aggregate Children in an FML Hierarchy Request

```
TABLE FILE filename
{PRINT|SUM} ....
FOR hierarchyfld
parentvalue ADD [n|ALL] [AS CAPTION|'text'] [LABEL label]
.
.
.
END
```

where:

filename

Is the name of the file to be used in the FML request. (If the hierarchy for this request cannot be loaded automatically, you must have previously issued the LOAD CHART command to load the hierarchy.)

hierarchyfld

Is the hierarchy field name. If the request references a joined structure, the name must be the field name from the host file. The alias name is not supported.

parentvalue

Is the parent value for which the children are to be aggregated.

n|ALL

Is a positive integer from 1 to 99 specifying the number of levels of the hierarchy to retrieve and aggregate. If *n* is omitted, direct children are retrieved. If *n* is 2, direct children and grandchildren are retrieved. ADD 99 retrieves children to 99 levels. ALL is a synonym for 99.

ADD

Displays all children on one row, summing the numeric data values displayed on the row. (This corresponds to the FML syntax CHILD1 OR CHILD2 OR ...)

CAPTION

Indicates that the caption of the parent value will display for the total row.

Note: The AS CAPTION phrase is supported for tagged rows that do not use the GET CHILDREN or ADD syntax. However, the hierarchy must be defined (by specifying the PARENT_OF attribute) in order to load and display the caption values. If the hierarchy is not defined, the AS CAPTION phrase is ignored.

'text'

Is a text string to use as the caption for the aggregate row. The CAPTION field defined in the Master File will not be used as the caption on the report output.

label

Is an explicit row label. Each generated row is labeled with the specified label text.

Example **Displaying an FML Hierarchy**

The following request displays two levels of account numbers below account 3000:

```
TABLE FILE CENTGL
PRINT GL_ACCOUNT_LEVEL
FOR GL_ACCOUNT
3000 OVER
3000 GET CHILDREN 2
END
```

The output is:

	Lev

3000	2
3100	3
3110	4
3120	4
3130	4
3140	4
3200	3
3300	4
3400	4
3500	4
3600	4
3700	4
3800	4
3900	4

Example **Displaying an FML Hierarchy With Captions**

The following request displays two levels of hierarchy below account 1000 (the top of the hierarchy) and displays the caption field values instead of the account numbers:

```
TABLE FILE CENTGL
PRINT  GL_ACCOUNT_LEVEL
FOR    GL_ACCOUNT
1000 AS CAPTION  OVER
1000 GET CHILDREN 2 AS CAPTION
END
```

The output is:

	Lev

Profit Before Tax	1
Gross Margin	2
Sales Revenue	3
Cost Of Goods Sold	3
Total Operating Expenses	2
Selling Expenses	3
General + Admin Expenses	3
Total R+D Costs	2
Salaries	3
Misc. Equipment	3

Example **Displaying an FML Hierarchy Summary Row**

The CENTSTMT data source is a rolled up data source. That is, each parent account is pre-populated with the sum of its direct children. Producing a sum of more than one level of child would add these intermediate totals into the summary, adding each child account multiple times. To avoid the double counting, this request adds only direct children.

All of the amounts in the data source are positive values, but some of them should be subtracted rather than added. The field GL_ROLLUP_OP contains a plus sign or minus sign to indicate which should be added and which should be subtracted.

The following request first prints the detail row for account 2000, then all of its children, and then a summary row for its direct children. Note that the ADD option requires the setting FORMULTIPLE=ON in order to use the same account record when constructing the detail rows and the summary row. The DEFINE command creates a field with the value of ACTUAL_AMT but with the sign indicated by the GL_ROLLUP_OP field:

```
SET FORMULTIPLE = ON
DEFINE FILE CENTSTMT
SIGNED_ACTUAL/D10.0 = IF GL_ROLLUP_OP EQ '+' THEN ACTUAL_AMT
                     ELSE  -ACTUAL_AMT;
END
TABLE FILE CENTSTMT
SUM SIGNED_ACTUAL
FOR GL_ACCOUNT
2000                AS CAPTION OVER
BAR                OVER
2000 GET CHILDREN ALL AS CAPTION OVER
BAR                OVER
2000 ADD            AS CAPTION
END
```

FML Hierarchies

The output is:

	SIGNED_ACTUAL
Gross Margin	25,639,225.
Sales Revenue	62,362,490.
Retail Sales	49,355,185.
Retail - Television	7,971,992.
Retail - Stereo	3,063,729.
Retail - Video Player	6,332,185.
Retail - Computer	1,204,955.
Retail - Video Camera	30,782,323.
Mail Order Sales	6,899,415.
Mail Order - Television	30,892.
Mail Order - Stereo	122,418.
Mail Order - Video Player	122,454.
Mail Order - Computer	2,427,478.
Mail Order - Video Camera	4,196,174.
Internet Sales	6,107,891.
Internet - Television	30,513.
Internet - Stereo	298,051.
Internet - Video Player	310,499.
Internet - Computer	1,204,687.
Internet - Video Camera	4,264,140.
Cost Of Goods Sold	-36,723,265.
Variable Material Costs	27,438,623.
Television COGS	4,927,162.
Stereo COGS	1,180,485.
Video COGS	3,644,631.
Computer COGS	2,395,093.
Video Camera COGS	15,291,254.
Direct Labor	6,176,900.
Fixed Costs	3,107,742.
Gross Margin	25,639,225.

The following variation of the request displays the direct children of account 2000 and a summary row for each of the first two periods. The caption for the summary lines has been replaced by the text 'TOTAL' using the AS phrase:

```

SET FORMULTIPLE = ON
DEFINE FILE CENTSTMT
SIGNED_ACTUAL/D10.0 = IF GL_ROLLUP_OP EQ '+' THEN ACTUAL_AMT
                        ELSE -ACTUAL_AMT;
END
TABLE FILE CENTSTMT
SUM SIGNED_ACTUAL
BY PERIOD
WHERE PERIOD LT '2002/03'
FOR GL_ACCOUNT
2000          AS CAPTION      OVER
BAR
2000 GET CHILDREN AS CAPTION OVER
BAR
2000 ADD      AS 'TOTAL'     OVER
" "
END

```

The output is:

PERIOD		SIGNED_ACTUAL
-----		-----
2002/01	Gross Margin	4,309,178.

	Sales Revenue	10,405,742.
	Cost Of Goods Sold	-6,096,564.

	TOTAL	4,309,178.
2002/02	Gross Margin	4,257,712.

	Sales Revenue	10,073,499.
	Cost Of Goods Sold	-5,815,787.

	TOTAL	4,257,712.

Reference Usage Notes for FML Hierarchies

- PROPERTY and REFERENCE are propagated to HOLD Master Files.
- The following setting is required in order to use a data record in more than one row of an FML request (for example, both a detail and summary row):

```
SET FORMULTOIPLE = ON
```

When reporting against a rolled up data source such as ESSBASE, the data values stored for the parent instance is an aggregate of all of its children. If you screen out any children using a WHERE phrase, the data displayed as a result of ADD will not aggregate to the data stored in the parent instance.

When reporting against a data source with shared members (such as ESSBASE), in which the same data can be defined multiple times with different hierarchy field values, data shared by two different parents will be counted twice in an aggregation operation. To avoid this double aggregation, use the FST. operator in the SUM command for the shared fields.

Reference FML Hierarchy Error Messages

LOAD CHART messages:

```
(FOC36361) REQUESTED HIERARCHY NOT FOUND FOR MASTER.
```

This message occurs at LOAD CHART time when the Master File cannot be found, or the hierarchy within the Master cannot be found.

```
(FOC36362) NOT ENOUGH MEMORY TO LOAD REQUESTED HIERARCHY.
```

This message occurs when there is not enough available memory to load the requested hierarchy.

```
(FOC36363) NO DATA FOUND FOR REQUESTED HIERARCHY.
```

The hierarchy does not have any data associated with it.

TABLE request messages:

```
(FOC36364) REQUESTED HIERARCHY IS NOT LOADED.
```

This message occurs when a hierarchy is needed to satisfy the request, but that hierarchy is not loaded.

```
(FOC36365) SYNTAX ERROR IN GET CHILDREN COMMAND
```

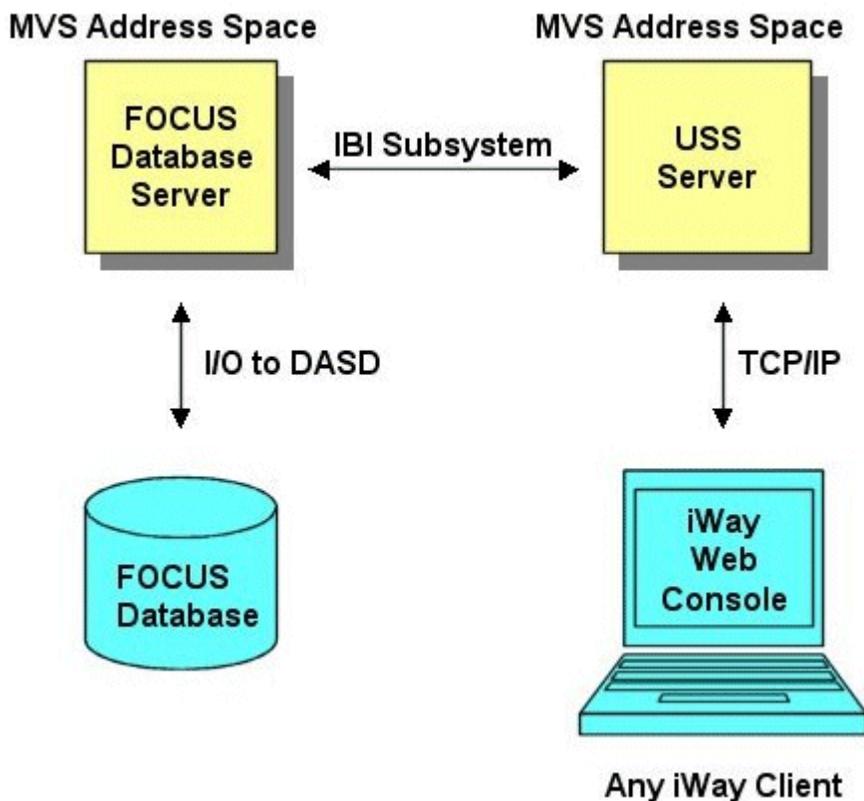
This message occurs when there is any kind of syntax error in the GET commands.

```
(FOC36366) WRONG NUMERIC VALUE IN GET CHILD n PHRASE.
```

Numeric value should be a positive integer.

FOCSBS

The IBI Subsystem now enables an iWay Server running under UNIX Systems Services (USS) to communicate with one or more FOCUS Database Servers (sink machines) running in the MVS environment:



This feature is supported in OS/390 2.10 and higher operating systems. It is not supported on VM operating systems.

Accessing a FOCUS Database Server on MVS

An iWay USS Server can access an unlimited number of FOCUS Database Servers running under MVS in addition to one FOCUS Database Server running under the iWay USS Server.

You enable communication between the USS Server and a FOCUS Database Server on MVS by adding an SBS node block to the USS Server's ODIN.CFG.

You must also issue a USE command for each FOCUS data source to be accessed on a FOCUS Database Server. You can include the USE commands in any supported profile on the USS Server or in a remote procedure call (RPC).

Syntax How to Access a FOCUS Database Server on MVS

1. Add an SBS block for each MVS FOCUS Database Server in the ODIN.CFG file on the USS Server:

```
; FOCUS Database Client
NODE = sbsname
BEGIN
  PROTOCOL = SBS
  CLASS = SUCLIENT
  SERVICE = qualif.focsu.data
END
```

where:

sbsname

Is a unique node name. This is also the ddname for the FOCUS Database Server communication data set. The server automatically performs the allocation of the communication data set based on the information in this configuration block. (The FOCUS Database Server startup JCL must allocate the same data set to ddname FOCSU.) This is the SU name that must be specified in the USE command needed to access a data source on the FOCUS Database Server.

Note: The ODIN.CFG file is created with two blocks named FOCSU and FOCSU01 during installation of the USS Server. These blocks provide access to the FOCUS Database Server running under the iWay USS Server. You cannot use either of these two names for this new block.

qualif.focsu.data

Is the fully-qualified name of the MVS FOCUS Database Server communication data set.

2. Issue a USE command for each FOCUS data source to be accessed on the FOCUS Database Server. You can issue the USE command in any supported profile on the USS Server or in an RPC:

```
USE focusdb1 ON sbsname
USE focusdb2 ON sbsname
```

Example Retrieving Data From a FOCUS Database Server on MVS

The ODIN.CFG file in the iWay Server on USS contains the following blocks. The first two are created automatically. The third block has been added for communicating to a FOCUS Database Server on MVS:

```

; FOCUS Database Server
NODE = FOCSU
BEGIN
PROTOCOL = TCP
CLASS = SUSERVER
SERVICE = 3562
END

; FOCUS Database Client
NODE = FOCSU01
BEGIN
PROTOCOL = TCP
CLASS = SUCLIENT
HOST = localhost
SERVICE = 3562
END

; FOCUS Database Client against MVS Sink Machine
NODE = FOCSBS
BEGIN
PROTOCOL = SBS
CLASS = SUCLIENT
SERVICE = USER1.FOCSU.DATA
END

```

The following REQUEST has a USE command that directs retrieval to the FOCUS Database Server on MVS.

```

USE GGSALES ON FOCSBS
END
SET PAGE = NOPAGE
TABLE FILE GGSALES
SUM UNITS DOLLARS
BY REGION BY CATEGORY
END

```

The output is:

Region	Category	Unit Sales	Dollar Sales
Midwest	Coffee	332777	4178513
	Food	341414	4338271
	Gifts	230854	2883881
Northeast	Coffee	335778	4164017
	Food	353368	4380004
	Gifts	227529	2848289
Southeast	Coffee	350948	4415408
	Food	349829	4308731
	Gifts	234455	2986240
West	Coffee	356763	4473527
	Food	340234	4202338
	Gifts	235042	2977092

Reference **FOCSBS Error Messages**

For the purpose of continuity, the error messages are similar to the messages generated by the MVS Server, but they are generated in response to new situations.

```
(FOC542)  SU. COMMUNICATION IS NOT AVAILABLE TO CENTRAL DATABASE  
          MACHINE: mfdname BYPASSING TO END OF COMMAND
```

This message is produced under any of the following conditions:

- IBI Subsystem is not operational.
- SBS Block is not defined in ODIN.CFG.
- FOCUS SU is not operational.

```
(FOC205)  THE DESCRIPTION CANNOT BE FOUND FOR FILE NAMED: mfdname
```

This message is produced if the USE command is not issued prior to a FOCUS request for data on the FOCUS Database Server.

```
(FOC036)  NO DATA FOUND FOR THE FOCUS FILE NAMED: command
```

This message is produced for each REQUEST line following the USE command if the USE command is not terminated by an END command.

HNODATA

The SET HNODATA command controls the missing data characters that are propagated to fields with the MISSING=ON attribute in HOLD FORMAT ALPHA files. Missing values in fields that do not have the MISSING=ON attribute are propagated to a HOLD file as blank (for alphanumeric fields) or zero (for numeric fields).

Syntax **How to Control Missing Values Propagated to a HOLD File**

`SET HNODATA = charstring`

In a TABLE request:

`ON TABLE SET HNODATA charstring`

where:

charstring

Is a string of up to 12 characters that will be propagated to a HOLD FORMAT ALPHA file for missing values in a field with the MISSING=ON attribute. The default value is a period (.).

If the string is longer than the length of the field, the value stored in:

- An alphanumeric field will be the leftmost characters of the string.
- A numeric field will be a blank string.

When an alphanumeric string other than the default value (the period) is used to populate a missing numeric field, a blank will be inserted in the held field to prevent a format error when displaying the data. If you use the default HNODATA value, it is inserted in numeric fields. In this way, a request against the HOLD file can recognize missing data that was propagated to the HOLD file.

If a number with decimal places is specified for HNODATA and the field with missing data is integer, the value will be rounded to a whole number and then inserted. In a numeric field that supports decimal places, it will be rounded and inserted with the correct number of decimal digits.

Example Controlling Missing Data Values in an Alphanumeric HOLD File

First edit the BANK_NAME and BANK_CODE fields in the EMPLOYEE Master File to support missing data:

```
FIELDNAME=BANK_NAME, ALIAS=BN, FORMAT=A20, MISSING=ON, $
```

```
FIELDNAME=BANK_CODE, ALIAS=BC, FORMAT=I6S, MISSING=ON, $
```

The following request sets the HNODATA string to the string 'N/A' and creates an alphanumeric HOLD file named HMISS:

```
SET HNODATA = 'N/A'
```

```
TABLE FILE EMPLOYEE
PRINT BANK_NAME BANK_CODE
BY EMP_ID
ON TABLE HOLD FORMAT ALPHA AS HMISS
END
-RUN
```

```
TABLE FILE HMISS
PRINT *
END
```

Note that a blank is inserted in the integer field BANK_CODE:

EMP_ID	BANK_NAME	BANK_CODE
-----	-----	-----
071382660	N/A	
112847612	N/A	
117593129	STATE	510271
119265415	N/A	
119329144	BEST BANK	302161
123764317	ASSOCIATED	112600
126724188	N/A	
219984371	N/A	
326179357	ASSOCIATED	112600
451123478	ASSOCIATED	112600
543729165	N/A	
818692173	BANK ASSOCIATION	175963

Example Propagating Numeric Values for Missing Data to a HOLD File

```

SET HNODATA = 123.65
TABLE FILE EMPLOYEE
PRINT BANK_NAME BANK_CODE
BY EMP_ID
ON TABLE HOLD FORMAT ALPHA AS HMISS
END
-RUN

```

```

TABLE FILE HMISS
PRINT *
END

```

The number 123.65 is rounded to 124 before being inserted into the integer BANK_CODE field:

EMP_ID	BANK_NAME	BANK_CODE
-----	-----	-----
071382660	123.65	124
112847612	123.65	124
117593129	STATE	510271
119265415	123.65	124
119329144	BEST BANK	302161
123764317	ASSOCIATED	112600
126724188	123.65	124
219984371	123.65	124
326179357	ASSOCIATED	112600
451123478	ASSOCIATED	112600
543729165	123.65	124
818692173	BANK ASSOCIATION	175963

Holding Missing Values

In HOLD files you can now distinguish between missing data and default values of blank (for character data) or zero (for numeric data) using the HOLDMISS command.

Syntax How to Store Missing Data in HOLD Files

```
SET HOLDMISS={ON|OFF}  
ON TABLE SET HOLDMISS {ON|OFF}
```

where:

ON

Allows you to store missing data in a HOLD file. When TABLE generates a default value for data not found, it generates missing values.

OFF

Does not allow you to store missing data in a HOLD file. OFF is the default.

Reference Usage Notes for Holding Missing Values

- Setting HOLDMISS ON adds the MISSING=ON attribute to every field in the extract file.
- Data will not be found if:
 - ALL is set to ON
 - The request is multi-path
 - An ACROSS statement has been issued

Example Holding Missing Values Using HOLDMISS

```
SET HOLDMISS=ON
TABLE FILE MOVIES
SUM WHOLESALEPR
BY CATEGORY ACROSS RATING
ON TABLE HOLD AS HLDM
END
TABLE FILE HLDM
PRINT *
END
```

The output is:

CATEGORY	WHOLESALEPR	WHOLESALEPR	WHOLESALEPR	WHOLESALEPR	WHOLESALEPR
ACTION	.	.	20.98	.	34.48
CHILDREN	54.49	51.38	.	.	.
CLASSIC	40.99	160.80	.	.	.
COMEDY	.	.	46.70	30.00	13.75
DRAMA	10.00
FOREIGN	13.25	.	62.00	.	70.99
MUSICALS	15.00	.	13.99	9.99	13.99
MYSTERY	.	9.00	18.00	9.00	80.97
SCI/FI	.	.	.	35.99	43.53
TRAIN/EX	.	60.98	.	.	.

Long Qualified Field Names

A long qualified field name can be comprised of a long file name, a long segment name, and a long field name. The file and segment names can be up to 64 characters in length while the field name can be up to 66 characters long.

Long qualified field names are useful for the storage of documents that use large descriptive field names such as those that are used and transmitted with the XML protocol.

Reference Usage Notes for Long Qualified Field Names

Long qualified field names can be used with MVS, UNIX, Windows NT, AS400, and USS.

The following is required in order to use long qualified field names:

- Create a Master File with a suffix of XFOC with long segment and field names.
- Allocate an XFOC data source data set. The DCB information for the data source is:
 - Record Format - FB
 - Logical Record Length (LRECL) - 16384
 - Block size - 16384
- Create a PDS library to contain wide FOCEXEC members. The DCB information for the library is:
 - Record Format - VB
 - Logical Record Length (LRECL) - 4096
 - Block size _ system defined DASD default

Note: PDS libraries containing wide FOCEXECs must always be concatenated in front of PDS libraries containing standard format FOCEXECs (fixed block with an LRECL of 80) in a CLIST, or in JCL.

- A wide FOCEXEC has to be coded with:
 - A DYNAM ALLOC command to access a data source containing a long file name.
 - A request using a long qualified field name.
 - Issue a CREATE FILE command against the XFOC data source.

Example How to Use Long Qualified Fieldnames

Sample Master File:

```
FILENAME=INTSALES, SUFFIX=XFOC  
SEGNAME=STORE_SALES_FROM_FRANCE, SEGTYPE=S1  
FIELDNAME=PROVINCE_NICE_STORE500_TOTALSALES, TSALE500, A10, FIELDTYPE=I, $  
FIELDNAME=PROVINCE_NICE_STORE501_TOTALSALES, TSALE501, A10, FIELDTYPE=I, $
```

Sample FOCEXEC:

Note: For display purposes, the FOCEXEC contains CREATE FILE and MODIFY commands.

```
DYNAM ALLOC DD INTSALES LONGNAME -  
ADVANT_GARDE_CLOTHING_INTERNATIONAL_SALES_DATASOURCE -  
DS PMSESD.INTSALES.FOCUS SHR REU  
  
-RUN  
  
- *  
  
CREATE FILE  
ADVANT_GARDE_CLOTHING_INTERNATIONAL_SALES_DATASOURCE  
  
-RUN  
  
- *  
  
MODIFY  
FILE ADVANT_GARDE_CLOTHING_INTERNATIONAL_SALES_DATASOURCE  
FREEFORM TSALE500 TSALE501  
MATCH TSALE500  
ON MATCH REJECT  
ON NOMATCH INCLUDE  
DATA  
12235,5470,$  
9827,7675,$  
7635,9327,$  
10325,7543,$  
END  
  
-RUN  
  
TABLE FILE ADVANT_GARDE_CLOTHING_INTERNATIONAL_SALES_DATASOURCE  
PRINT  
ADVANT_GARDE_CLOTHING_INTERNATIONAL_SALES_DATASOURCE.STORE_SALES_FROM_FRA  
NCE.PROVINCE_NICE_STORE500_TOTALSALES  
ADVANT_GARDE_CLOTHING_INTERNATIONAL_SALES_DATASOURCE.STORE_SALES_FROM_FRA  
NCE.PROVINCE_NICE_STORE501_TOTALSALES  
END
```

-RUN

Sample output from FOCEXEC:

PROVINCE_NICE_STORE500_TOTALSALES	PROVINCE_NICE_STORE501_TOTALSALES
-----	-----
12235	5470
9827	7675
7635	9327
10325	7543

Reference Error Messages for Long Qualified Field Names

The following error message is displayed when either:

- A long file name exceeds 64 characters in length.
- A long segment name exceeds 64 characters in length.
- A long field name exceeds 66 characters in length.
- A long qualified field name exceeds 194 characters in length.

```
(FOC003) THE FIELDNAME IS NOT RECOGNIZED: fieldname
        BYPASSING TO END OF COMMAND
```

Minus Edit Format Option

FOCUS provides a format option that enables you to display a minus sign to the right of negative numeric data. The option can be defined in a Master File as well as in a COMPUTE or DEFINE request.

Syntax How to Display Minus Signs

Format	Data	Display
I2-	-21	21-
D7-	-6148	6148-
F7.2-	-8878	8878.00-

Example Describing the Minus Edit Format Option

In the Master File:

FIELDNAME=SHIP_COST, SCOST, D7-, \$

In a REQUEST:

```

DEFINE FILE MOVIES
MDISC/D7.2 = 7.5;
END
TABLE FILE MOVIES
SUM COMPUTE GROSS_PROFIT = LISTPR - WHOLESALEPR;
NET_PROFIT/D12.2- = GROSS_PROFIT - MDISC;
BY CATEGORY
PRINT MDISC AS 'MEMBER,DISCOUNT'
WHOLESALEPR AS 'WHOLESALE,PRICE'
LISTPR AS 'LIST,PRICE'
IF CATEGORY EQ 'ACTION' OR 'MUSICALS' OR 'DRAMA' OR 'SCI/FI';
IF RECORDLIMIT EQ 10
END
    
```

The output is:

CATEGORY	GROSS_PROFIT	NET_PROFIT	MEMBER DISCOUNT	WHOLESALE PRICE	LIST PRICE
-----	-----	-----	-----	-----	-----
ACTION	8.96	1.46	7.50	10.99	19.95
DRAMA	9.98	2.48	7.50	10.00	19.98
MUSICALS	16.97	5.53-	7.50	13.99	19.98
			7.50	13.99	19.98
			7.50	9.99	14.98
SCI/FI	35.32	2.18-	7.50	14.99	19.98
			7.50	14.55	19.98
			7.50	13.99	19.95
			7.50	16.00	24.95
			7.50	19.99	29.98

Note: Results of the Minus Edit option can be seen in the Net_Profit column.

Reference Usage Notes for Minus Edit

- Minus Edit does not perform calculations intended to derive a negative number in the data. It is only a format option that allows numeric data to be displayed with a minus sign.
- A blank space is displayed to the right side of the data when the data is positive in value.
- Using the Minus Edit format option with format options B, E, R, T, DMY, MDY, and YMD results in the display of an error message.
- Placing a minus sign on the right side of data fields in a CRTFORM results in the display of an error message.

Reference Error Messages for Minus Display Option

Using the Minus Edit format option with format options B, E, R, T, DMY, MDY, and YMD results in the following error message being displayed:

(FOC253) INVALID FORMAT SPECIFICATION ON LEFT HAND SIDE: D12.2R-

Using the Minus Edit format option with two minus signs also results in the preceding error message being displayed.

Placing a minus sign on the right side of data fields in a CRTFORM results in the following error message being displayed:

(FOC210) THE DATA VALUE HAS A FORMAT ERROR:

An alphabetic character has been found where all numerical digits are required.

Multiple FOLD-LINE in a TABLE Request

You can now specify multiple FOLD-LINE clauses in a TABLE request. Up to sixteen instances of FOLD-LINE may be used in a TABLE request.

Syntax **How to Create Multiple FOLD-LINE Clauses in a TABLE Request**

```
display fieldname1... FOLD-LINE fieldname2...
```

```
{ON|BY} fieldname3 FOLD-LINE
```

where:

```
display
```

Is any display command.

```
fieldname2, fieldname3, ...
```

Because these field names are after the FOLD-LINE clause, the columns that come after them are placed on a separate line when the value of the field changes in the BY or ON phrase.

The field name may be a sort field or display field. If it is a display field, the next field is placed at the start of the next line. If it is a sort field, the next field is on a new line, offset by two spaces from the beginning of the current line.

Example Specifying Multiple FOLD-LINE Clauses in a TABLE Request

```
TABLE FILE MOVIES
SUM WHOLESALER LISTPR
BY CATEGORY FOLD-LINE
BY TITLE
BY DIRECTOR FOLD-LINE
IF CATEGORY EQ 'FOREIGN' OR 'DRAMA'
END
```

The output is:

```
CATEGORY
  TITLE                                DIRECTOR
    WHOLESALER LISTPR
-----
DRAMA
  DOG DAY AFTERNOON                    LUMET S.
    10.00    19.98
FOREIGN
  BABETTE'S FEAST                      AXEL G.
    13.25    19.98
  DAS BOOT                              PETERSON W.
    15.99    29.95
  DEATH IN VENICE                      VISCONTI L.
    30.00    59.95
  FAMILY, THE                          SCOLA E.
    32.00    59.95
  FANNY AND ALEXANDER                  BERGMAN I.
    25.00    39.98
  TIN DRUM, THE                        SCHLONDORFF V.
    30.00    59.95
```

Reference Usage Notes for Compressing Report Columns

- Full compatibility with a single FOLD-LINE is preserved.
- Up to 16 instances of FOLD-LINE can be used in a table request in total.
- There will be no '-----' after each header in case if there is more than one fold. This helps to make headers more compact.

Multiple Repeated Values in FOR Statement

An input record can now be displayed and used in a calculation within multiple FOR rows in FML.

Syntax **How to Display the Same Record in Multiple FOR Rows**

`SET FORMULTIPLE= {ON | OFF}`

`ON TABLE SET FORMULTIPLE= {ON | OFF}`

where:

`ON`

Allows you to include the same record in multiple FOR rows.

`OFF`

Does not allow you to include the same record in multiple FOR rows. If there are multiple FOR rows present in an FML request for the same record, and FORMULTIPLE is off, the record will only be included in one of the FOR rows. OFF is the default.

Example **Displaying the Same Record in Multiple FOR Rows**

```

SET FORMULTIPLE = ON
TABLE FILE LEDGER
SUM AMOUNT
FOR ACCOUNT
1010 AS 'CASH ON HAND' OVER
1020 AS 'DEMAND DEPOSITS' OVER
1030 AS 'TIME DEPOSITS' OVER
BAR OVER
1010 OR 1020 OR 1030 AS 'TOTAL CASH' OVER
" " OVER
1100 AS 'ACCOUNTS RECEIVABLE' OVER
1200 AS 'INVENTORY' OVER
BAR OVER
1100 TO 1200 AS 'TOTAL NON-CASH ASSETS'
END

```

The output is:

	AMOUNT

CASH ON HAND	8,784
DEMAND DEPOSITS	4,494
TIME DEPOSITS	7,961

TOTAL CASH	21,239
ACCOUNTS RECEIVABLE	18,829
INVENTORY	27,307

TOTAL NON-CASH ASSETS	46,136

Note: The accounts are referenced in more than one row of the request.

Multi-Variate Regress

The REGRESS method derives a linear equation that best fits a set of numeric data points and uses this equation to create a new column in the report output. The equation can be based on one to three independent variables

Note: This feature has Candidate for Release status in iWay Version 5.2.0.

This method estimates values by assuming that the dependent variable (y, the new calculated values) and the independent variables (x1, x2, x3) are related by the following linear equation:

$$y = a1*x1 [+ a2*x2 [+ a3*x3]] + b$$

When there is one independent variable, the equation represents a straight line. This produces the same values as FORECAST using the REGRESS method. When there are two independent variables, the equation represents a plane, and with three independent variables, it represents a hyperplane. You should use this technique when you have reason to believe that the dependent variable can be approximated by a linear combination of the independent variables.

REGRESS uses a technique called Ordinary Least Squares to calculate values for the coefficients (a1, a2, a3, and b) that minimize the sum of the squared differences between the data and the resulting line, plane, or hyperplane.

Syntax

How to Create a Multivariate Linear Regression Column

```
ON {sortfield|TABLE} RECAP y[/fmt] = REGRESS(n, x1, [x2, [x3,]] x);
```

where:

sortfield

Is a field in the data source. It cannot be the same field as any of the parameters to REGRESS. A new linear regression equation is derived each time the sort field value changes.

y

Is the new numeric column calculated by applying the regression equation. You cannot DEFINE or COMPUTE a field with this name.

fmt

Is the display format for y. If it is omitted, the default format is D12.2.

n

Is a whole number from 1 to 3 indicating the number of independent variables.

x1, x2, x3

Are the field names to be used as the independent variables. All of these variables must be numeric and be independent of each other.

z

Is an existing numeric field that is assumed to be approximately linearly dependent on the independent variables and is used to derive the regression equation.

Reference Usage Notes for REGRESS

- The sort field used for REGRESS must be a numeric or smart date field. REGRESS cannot operate on an ACROSS field. If any of the independent variables are also sort fields, they cannot be referenced in the request prior to the REGRESS sort field.
- The SUM command is not supported with REGRESS.
- FORECAST and REGRESS cannot be used in the same request, and only one REGRESS is supported in a request. Non-REGRESS RECAP commands are supported.
- The RECAP command used with REGRESS can contain only the REGRESS syntax. REGRESS does not recognize any syntax after the closing semicolon (;).
- Although you pass parameters to REGRESS using an argument list in parentheses, REGRESS is not a function. It can coexist with a user-written subroutine of the same name, as long as the user-written subroutine is not specified in a RECAP command.
- BY TOTAL is not supported.
- MORE, MATCH, FOR, and OVER are not supported.
- The process of generating the REGRESS values creates extra columns that are not printed in the report output. The number and placement of these additional columns varies depending on the specific request. Therefore, use of column notation is not supported in a request that includes REGRESS.
- SUMMARIZE and RECOMPUTE are not supported for the same sort field used for REGRESS.
- REGRESS is not supported for the FOCUS GRAPH facility; it is supported for the iWay GRAPH facility.
- The left side of a RECAP command used for REGRESS supports the CURR attribute for creating a currency denominated field.
- Fields with missing values cannot be used in the regression.

Example **Creating a Multivariate Linear Regression Column**

The following request uses the GGSALES data source to calculate an estimated DOLLARS column. The BUDUNITS, UNITS, and BUDDOLLARS fields are the independent variables. The DOLLARS field provides the actual values to be estimated:

```

DEFINE FILE GGSALES
SDATE/YYM = DATE;
SYEAR/Y = SDATE;
END

TABLE FILE GGSALES
PRINT BUDUNITS UNITS BUDDOLLARS DOLLARS
ON TABLE RECAP ESTDOLLARS/F8 = REGRESS(3, BUDUNITS,
UNITS, BUDDOLLARS, DOLLARS);
WHERE SYEAR EQ 97
WHERE CATEGORY EQ 'Coffee'
WHERE REGION EQ 'West'
WHERE UNITS GT 1600 AND UNITS LT 1700
END

```

The output is:

Budget Units	Unit Sales	Budget Dollars	Dollar Sales	ESTDOLLARS
1796	1696	17960	25440	23381
1463	1663	21945	19966	18919
1759	1615	17590	17765	20850
1403	1601	21045	17611	16530
1825	1695	23725	25425	24316
1613	1685	22582	18535	21365
1568	1682	23520	25230	20841
1733	1647	17330	24705	21313
1870	1620	24310	24300	23033
1665	1678	21645	23492	21722
1725	1669	22425	21697	22307
1457	1671	21855	20052	19038
1662	1674	24930	18414	21931
1556	1643	18672	18073	19201
1653	1694	21489	16940	21963
1473	1616	14730	16160	17095

Reference REGRESS Error Messages**(FOC36323) EXPRESSIONS ARE NOT PERMITTED IN PARAMETERS**

In FORECAST/REGRESS, parameters cannot be presented as expressions but only as fields or constants.

(FOC36324) VERB OBJECT OR INTERVAL CANNOT BE A QUOTED CONSTANT

All parameters of FORECAST/REGRESS except 'method' should not contain quotes.

(FOC36325) FORECAST/REGRESS VERB OBJECT IS NOT RECOGNIZED

The verb object is not found in either a Master File or a prior DEFINE section or a COMPUTE statement.

(FOC36330) INTERNAL FORECAST/REGRESS ERROR

Unforeseen error has occurred in routines for FORECAST/REGRESS. Please report the problem to an Information Builders representative.

(FOC36333) INVALID FORECAST/REGRESS REQUEST

'MORE', 'OVER', 'GRAPH', 'MODIFY', 'FOR', and 'BY TOTAL' as well as re-formatting into non-numeric type of field are not permitted in FORECAST/REGRESS.

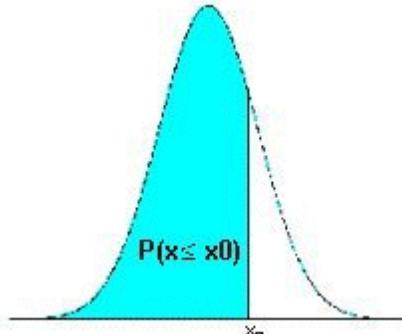
(FOC36341) INCORRECT USE OF PARAMETERS FOR REGRESSION

The presence of keys equal or preceding the 'ON field' in regression parameter list as well as reusing parameters in the list is not allowed.

New Functions **NORMSDST** and **NORMSINV**

The *NORMSDST* subroutine calculates the cumulative standard normal distribution function.

Given a point x_0 , *NORMSDST* returns a value that represents the area to the left of point x_0 under the standard normal distribution curve. This area corresponds to the probability that an arbitrary point, x , is less than or equal to x_0 :



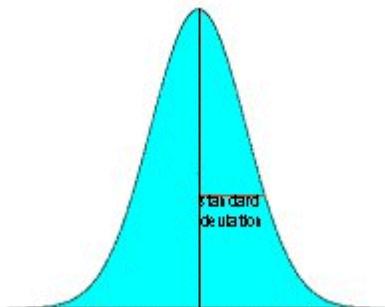
The value returned by *NORMSDST* also represents the percentage of points that are less than or equal to x_0 , making this subroutine useful for determining percentiles of normally distributed data.

Given an area or probability on the standard normal curve, the *NORMSINV* subroutine returns the x value that corresponds to the upper boundary of that area.

NORMSINV is the inverse of *NORMSDST*. That is, $\text{NORMSINV}(\text{NORMSDST}(x)) = x$.

Characteristics of the Normal Distribution

Many common measurements tend to be normally distributed. A plot of normally distributed data values approximates a bell-shaped curve:



The two measures needed to describe any normal distribution are the mean and the standard deviation:

- The mean is the point at the center of the curve.
- The standard deviation describes the spread of the curve. It is the distance from the mean to the point of inflection (point where the curve changes direction).

The standard normal distribution is a normal distribution that has a mean of 0 and a standard deviation of 1. The total area under this curve is 1. A point on the x-axis of the standard normal distribution is called a z-score or normalized value. Assuming that your data is normally distributed, you can convert a data point (raw score) to a z-score in order to find the percentage of scores that are less than or equal to the raw score.

If you use the z-score as input to the NORMSDST function, the value returned is the percent of scores less than or equal to the z-score (probability that a score is less than or equal to the z-score).

The inverse of the NORMSDST subroutine is called NORMSINV. It takes a percent of the curve and returns the z-score that marks the upper boundary of that portion of the curve.

Procedure **How to Convert a Value to a Z-Score**

You can convert a value (raw score) from your normally distributed data to the equivalent point on the standard normal distribution (z-score) as follows:

$$z = (\text{raw_score} - \text{mean}) / \text{standard_deviation}$$

To convert from a z-score back to a raw score, use the following formula:

$$\text{raw_score} = z * \text{standard_deviation} + \text{mean}$$

where:

z

Is the normalized data point, called a z-score.

raw_score

Is the raw (actual) data value.

Syntax **How to Calculate the Mean of a Set of Values**

The mean of data points x_i , where i is from 1 to n is:

$$(\sum x_i) / n$$

Syntax **How to Calculate the Standard Deviation of a Set of Values**

The standard deviation of data points x_i , where i is from 1 to n is:

$$\text{SQRT}[(\sum x_i^2 - (\sum x_i)^2 / n) / (n - 1)]$$

Syntax **How to Calculate the Cumulative Standard Normal Distribution Function**

$$\text{NORMSDST}(z, 'D8');$$

where:

z

Is a z-score (x value from a standard normal distribution).

The value returned by the subroutine is double precision. You can assign it to a field with any valid numeric format.

Syntax **How to Calculate the Inverse Cumulative Standard Normal Distribution Function**

```
NORMSINV (p, 'D8');
```

where:

p

Is a number between 0 and +1 (cumulative probability from a standard normal distribution).

The value returned by the subroutine is double precision. You can assign it to a field with any valid numeric format.

Example **Using the NORMSDST and NORMSINV Subroutines**

This request uses the GGPRODS data source. This data source contains a very small set of points which cannot be assumed to be normally distributed and are used only for demonstration purposes.

The field used is SIZE, which is converted to double precision (as X) for use with the AVE. and ASQ. prefix operators. Because the SIZE values are not normalized, a two-part request is required:

- The SUM command calculates the mean and standard deviation.
- The PRINT command uses the mean and standard deviation to convert the field values to normalized scores for input to NORMSDST.

- The NORMSINV subroutine is used to return the original z-score. The mean and standard deviation are used to convert the z-score back to a SIZE value:

```

DEFINE FILE GGPRODS
  -* CREATE AN APPROPRIATE NUMERIC SORT FIELD
  PRODNO/I4 = DECODE PRODUCT_ID('B141' 45 'B142' 42 'B144' 44
  'F101' 101 'F102' 102 'F103' 103 'G100' 140
  'G104' 144 'G110' 150 'G121' 151);
  -* CONVERT SIZE FIELD TO DOUBLE PRECISION
  X/D12.5 = SIZE;
  END

  TABLE FILE GGPRODS
  SUM X NOPRINT CNT.X NOPRINT
  -* CALCULATE MEAN AND STANDARD DEVIATION
  COMPUTE N0/D12.5 = CNT.X; NOPRINT
  COMPUTE MEAN/D12.5 = AVE.X; NOPRINT
  COMPUTE VARIANCE/D12.5 = ((N0*ASQ.X) - (X*X/N0))/(N0-1); NOPRINT
  COMPUTE STDEV/D12.5 = SQRT(VARIANCE); NOPRINT

  PRINT SIZE X NOPRINT
  -* COMPUTE Z-SCORES AND USE AS INPUT TO NORMSDST FUNCTION
  -* THEN USE RETURNED VALUES AS INPUT TO NORMSINV FUNCTION
  -* AND CONVERT BACK TO DATA VALUES
  COMPUTE Z/D12.5 = (X - MEAN)/STDEV;
  COMPUTE NORMSD/D12.5 = NORMSDST(Z, 'D8');
  COMPUTE NORMSI/D12.5 = NORMSINV(NORMSD, 'D8');
  COMPUTE DSIZE/D12 = NORMSI * STDEV + MEAN;
  BY PRODNO NOPRINT
  END
  
```

The output is:

Size	Z	NORMSD	NORMSI	DSIZE
12	-.80273	.21106	-.80273	12
12	-.80273	.21106	-.80273	12
16	-.07298	.47091	-.07298	16
20	.65678	.74434	.65678	20
24	1.38654	.91721	1.38654	24
20	.65678	.74434	.65678	20
24	1.38654	.91721	1.38654	24
16	-.07298	.47091	-.07298	16
12	-.80273	.21106	-.80273	12
8	-1.53249	.06270	-1.53249	8

Reference Usage Notes for the NORMSDST and NORMSINV Subroutines

- Data must come from a normal distribution.
- Results are accurate to six significant digits.
- NORMSINV uses an iterative technique. If it does not converge after 100 iterations, the function returns a zero value.
- Return values are generated as double precision. They can be assigned to F, P, or D formatted fields only.

SET CDN Command Enhancement

Continental Decimal Notation (CDN) includes the SPACE and QUOTE settings to support countries that require spaces and apostrophes in numbers.

Syntax How to Set Continental Decimal Notation

```
SET CDN = {ON|OFF|SPACE|QUOTE}
```

where:

ON

Sets the decimal point as a comma and the thousands separator as a period. ON should be used for Germany, Denmark, Italy, Spain, and Brazil.

OFF

Sets the decimal point as a period and the thousands separator as a comma. OFF is the default.

OFF should be used for the USA, Canada, Mexico, and the United Kingdom.

SPACE

Sets the decimal point as a comma and the thousands separator as a space. SPACE should be used for France, Norway, Sweden, and Finland.

QUOTE

Sets the decimal point as a comma and the thousands separator as an apostrophe. QUOTE should be used for Switzerland.

Example **How Continental Decimal Notation is Displayed**

The following table shows how 1234.56 is displayed depending on the setting of CDN.

OFF	1,234.56
ON	1.234,56
SPACE	1 234,56
QUOTE	1' 234,56

SET SAVEDMASTERS

Up to 99 Master Files can be saved in memory after they have been used in a request. Saved Master Files are not re-parsed when referenced in subsequent requests, producing significant performance improvement. The most recently used Master File is stored in memory regardless of this setting. Therefore, this technique is helpful when you run multiple requests against several Master Files.

Note: This feature has Candidate for Release status in iWay 5.2.0.

Syntax **How to Save Parsed Master Files in Memory**

`SET SAVEDMASTERS = n`

where:

n

Is an integer between 0 and 99 that specifies the maximum number of Master Files on the SAVEDMASTERS list. The default value is 0. Note that the most recently used Master File is always stored in memory, even with SAVEDMASTERS set to zero. However, the zero setting does not generate the list of saved Master Files.

Reference Usage Notes for SET SAVEDMASTERS

- With each request that specifies a new Master File, the prior Master File is moved down on the saved list and the new Master File is placed at the top of the list. Once all of the slots on the list are full, parsing a new Master File causes the one at the bottom to drop off the list.
- If an already saved Master File is used in a request, it moves to the top of the list.
- Only one occurrence of a Master File name is maintained on the list. Therefore, if you use an already saved Master File as the host file in a JOIN, in a HOLD command (with the same AS name), in a USE...AS command, or in a COMBINE command without specifying a unique name, the new version of the Master File replaces the previous version on the list.
- Re-parsing or reusing a saved Master File moves the Master File name to the top of the SAVEDMASTERS list.
- Memory resources are used to store the parsed Master Files, reducing the amount of memory available for other processes.
- You cannot selectively purge Master Files from the list.
- The SAVEDMASTERS parameter is not supported in a request (ON TABLE SET).
- The SAVEDMASTERS setting is not supported on a FOCUS Database Server or with a Maintain procedure.
- Issuing the CHECK FILE or REBUILD command causes the Master File to be re-parsed.
- The ?F and ?FF commands only re-parse the Master File when issued outside of a request for a Master File other than the most recently used Master File.
- Using an alternate file view (TABLE FILE filename.fieldname) or the AUTOPATH=ON setting re-parses the Master File.
- If the SAVEDMASTERS value is changed between requests:
 - Raising the number allows more Master Files to be saved as they are parsed.
 - Lowering the number drops the oldest saved Master Files.
- If changes are made to a Master File that is saved, the changes will not be implemented until the Master File is re-parsed.
- A JOIN CLEAR or USE CLEAR command does not purge the parsed Master File from memory. You must make sure the Master File is re-parsed before reusing it.

SUFFIX = COM/COMT/TABT

Syntax **How to Query the SAVEDMASTERS Setting**

The following query command indicates the number of Master Files allowed on the list of saved Master Files and lists the names of the Master Files on the list.

```
? SET SAVEDMASTERS
```

Example **Saving and Querying Parsed Master Files**

The following command specifies that up to three parsed Master Files can be saved:

```
SET SAVEDMASTERS = 3  
? SET SAVEDMASTERS
```

The output of the query command indicates that the list can contain up to three Master Files, but none are currently saved:

```
SAVEDMASTERS 3
```

The following procedure parses two Master Files, EMPLOYEE and MOVIES:

```
TABLE FILE EMPLOYEE  
PRINT LAST_NAME FIRST_NAME BY EMP_ID  
END  
-RUN  
TABLE FILE MOVIES  
PRINT TITLE BY DIRECTOR  
END  
-RUN  
? SET SAVEDMASTERS
```

The output of the query command indicates that the list can contain up to three Master Files and that the list currently consists of MOVIES and EMPLOYEE:

```
SAVEDMASTERS 3  
  
MOVIES  
EMPLOYEE
```

SUFFIX = COM/COMT/TABT

Master Files for comma and tab-delimited sequential data sources can have SUFFIX values of COM, COMT, or TABT. Comma-delimited data sources are sequential data sources in which field values are separated by commas. Tab-delimited data sources are sequential data sources in which field values are separated by tabs.

Note that comma-delimited and tab-delimited data sources cannot participate in joins.

Accessing SUFFIX=COM Data Sources

A Master File containing the attribute SUFFIX=COM can be used to access two styles of comma-delimited sequential data sources:

- One style is consistent with prior releases. Character values are not enclosed in double quotation marks, and the comma-dollar sign character combination is the record terminator. With this style of comma-delimited data source, records can span multiple lines. A field that contains a comma as a character must be enclosed within single quotation marks.
- The second style is consistent with the current industry standard for comma-delimited data sources. Character values are enclosed in double quotation marks and the CRLF (carriage-return, line-feed) character combination is the record terminator. In addition, each input record must be completely contained on a single input line. A double quotation mark within a field is identified by two consecutive double quotation marks.

Note that the setting PCOMMA=ON is required in conjunction with the SUFFIX=COM Master File when accessing this type of data source in order to correctly interpret the double quotation marks around character values. Without this setting, the double quotation marks are considered characters within the field, not delimiters enclosing the field values.

Accessing SUFFIX=COMT Data Sources

A Master File containing the attribute SUFFIX=COMT can be used to access comma-delimited sequential data sources in which all of the following conditions are met:

- The first record of the data source contains column titles. This record will be ignored when the data source is accessed in a request.
- Character values are enclosed in double quotation marks. A double quotation mark within a field is identified by two consecutive double quotation marks.
- Each record is completely contained on one line and terminated with the CRLF character combination.

Accessing SUFFIX=TABT Data Sources

A Master File containing the attribute SUFFIX=TABT can be used to access tab-delimited sequential data sources in which all of the following conditions are met:

- The first record of the data source contains column titles. This record will be ignored when the data source is accessed in a request.
- Character values are not enclosed in double quotation marks.
- Each record is completely contained on one line and terminated with the CRLF character combination.

Summary Prefix Operators

You can use the SUBTOTAL, SUB-TOTAL, RECOMPUTE, and SUMMARIZE commands at the ON TABLE level to specify the type of summary operation to use to produce the grand total line on the report.

In addition, prefix operators can be used with the summary options SUBTOTAL, SUB-TOTAL, RECOMPUTE, and SUMMARIZE at both the sort break grand total levels. Each type of summary has its own purpose and handles the prefix operators appropriately for the type of summary information to be displayed. For example, using AVE. at a sort field break produces the average within the sort group.

Prefix operations on summary lines are performed on the retrieved, selected, and summed values that will become the detail lines in the report. They are not performed on each incoming record as field-based prefix operations are.

Different operations from two ON phrases for the same sort break display on the same summary line and allow a mixture of operations on summary lines. The grand total line populates all fields populated by any summary command, even fields that are not specified in the grand total command. If the same field is referenced in more than one ON phrase for the same sort break, the last function specified is applied.

The following prefix operators are supported:

- ASQ.
- AVE.
- CNT.
- FST.
- LST.
- MAX.
- MIN.
- SUM.

Syntax **How to Use Prefix Operators on Summary Values**

At a sort break, use the following syntax

```
{BY|ON} breakfield [AS 'text1'] summaryoption [MULTILINES] operator_and_field_list
      [AS 'text2'] [WHEN expression;]
```

To replace the default grand total, use the following syntax

```
ON TABLE summaryoption operator_and_field_list [AS 'text2']
```

where:

breakfield

Is the sort field whose change in value triggers the summary operation.
summaryoption Can be one of the following: SUBTOTAL, SUB- TOTAL, RECOMPUTE, or SUMMARIZE.

'text1'

Is the column heading to use for the break field on the report output.

operator_and_field_list

Can have one of the following forms:

```
[field1 [field2 ... fieldn]]
```

Produces the type of summary specified by summaryoption for the listed fields. If no field names are listed, the summary is produced for every numeric column in the report output.

prefixop.

The prefix operator is applied to every numeric column in the report output. Every numeric column is populated with values on the summary row.

```
prefixop1. field1 [field2 ... fieldn] [prefixop2. fieldm ...]
```

The first prefix operator is applied to field1 through fieldn. The second prefix operator is applied to fieldm. Only the fields specified are populated with values on the summary row. Note that each prefix operator must be separated from the following field name by a blank space.

where:

```
prefixop1 ... prefixop2
```

Are prefix operators for fields used in summary operations. Note: Each prefix operator must be separated from the field name by a space.

```
field1 ... fieldm
```

Are the field names to be displayed for the summary line for the specified sort break. Each field name must be preceded by a space.

'text2'

Is the text that prints on the left of the summary row. expression Is an expression that determines whether the summary operation is performed at each break.

Reference Usage Notes for Summary Prefix Operators

COLUMN-TOTAL does not support prefix operators.

- A prefix operator on a summary line must always be followed by a blank space. For example:

```
ON RATING SUBTOTAL AVE. LISTPR
```
- Prefix operators PCT., RPCT., AND TOT. are not supported.
- Double prefix operators (such as PCT.CNT.) are not supported.
- The sort field that triggers a summary row cannot be an ACROSS field. An ACROSS field can be used in the request, in which case the same field name will display over multiple columns (ACROSS groups) in the report output. A prefix operator applied to such a field on a summary line is applied to all of those columns.
- The SUM. prefix operator produces the same summary values as a summary phrase with no prefix operator.
- SUMMARIZE and RECOMPUTE apply the calculations defined in the associated COMPUTE command to the summary values. Therefore, in order to perform the necessary calculations, the SUMMARIZE or RECOMPUTE command must calculate all of the fields referenced in the COMPUTE command.
- As in the case without prefix operators, different types of summary phrases cannot be combined in one report request.
- If the same field is referenced by more than one summary operation with different prefix operators at each level, the default grand total (one produced without an ON TABLE summaryoption command) applies the operation specified by the first operator used in the report request (left- most sort field in the output).

Example Using Prefix Operators With SUBTOTAL at the Sort Break Level

The following example uses prefix operators to calculate the:

- Average list price by rating within category.
- Sum of copies by category.

Notice that the subtotal row for each rating contains a value only in the LISTPR column and the subtotal row for each category contains a value only in the COPIES column. The default grand total line contains values only for the columns that were subtotaled. Note the blank space between each prefix operator and the following field name:

```
TABLE FILE MOVIES
PRINT COPIES LISTPR WHOLESALPR TITLE/A23
BY RATING BY CATEGORY
WHERE CATEGORY EQ 'CHILDREN' OR 'CLASSIC'
WHERE RATING EQ 'G' OR 'NR'
ON RATING SUBTOTAL AVE. LISTPR AS '*Ave: '
ON CATEGORY SUBTOTAL SUM. COPIES AS '*Sum: '
END
```

Summary Prefix Operators

The output is:

RATING	CATEGORY	COPIES	LISTPR	WHOLESALEPR	TITLE
-----	-----	-----	-----	-----	-----
G	CHILDREN	2	44.95	29.99	SHAGGY DOG, THE
		2	29.95	12.50	ALICE IN WONDERLAND
		3	26.99	12.00	BAMBI
*Sum:	CHILDREN	7			
	CLASSIC	3	89.95	40.99	GONE WITH THE WIND
*Sum:	CLASSIC	3			
*Ave:	G		47.96		
NR	CHILDREN	1	19.95	10.00	SMURFS, THE
		1	19.95	9.75	SCOOBY-DOO-A DOG IN THE
		1	14.95	7.65	SESAME STREET-BEDTIME S
		1	14.98	7.99	ROMPER ROOM-ASK MISS MO
		1	29.95	15.99	SLEEPING BEAUTY
*Sum:	CHILDREN	5			
	CLASSIC	1	24.98	14.99	EAST OF EDEN
		3	39.99	20.00	CITIZEN KANE
		1	29.95	15.99	CYRANO DE BERGERAC
		1	19.99	10.95	MARTY
		2	19.99	10.95	MALTESE FALCON, THE
		2	19.95	9.99	ON THE WATERFRONT
		2	89.99	40.99	MUTINY ON THE BOUNTY
		2	19.99	10.95	PHILADELPHIA STORY, THE
		2	19.98	10.99	CAT ON A HOT TIN ROOF
		2	29.95	15.00	CASABLANCA
*Sum:	CLASSIC	18			
*Ave:	NR		27.64		
TOTAL		33	31.91		

Example Using SUBTOTAL at the Sort Break and Grand Total Levels

The following example adds the ON TABLE SUBTOTAL command to the previous request to calculate the minimum number of copies and maximum list price on the grand total line for the entire report:

```
TABLE FILE MOVIES
PRINT COPIES LISTPR WHOLESALPR TITLE/A23
BY RATING
BY CATEGORY
WHERE CATEGORY EQ 'CHILDREN' OR 'CLASSIC'
WHERE RATING EQ 'G' OR 'NR'
ON RATING SUBTOTAL AVE. LISTPR AS '*Ave: '
ON CATEGORY SUBTOTAL SUM. COPIES AS '*Sum: '
ON TABLE SUBTOTAL MIN. COPIES MAX. LISTPR
END
```

The output is exactly the same as in the previous request except for the grand total line:

RATING	CATEGORY	COPIES	LISTPR	WHOLESALPR	TITLE
G	CHILDREN	2	44.95	29.99	SHAGGY DOG, THE
.					
.					
.					
TOTAL		1	89.99		

Controlling Summary Line Processing

Processing of reports that use prefix operators on summary lines differs from processing without prefix operators. In some cases, a different style of report output results from each type of request.

If a summary command specifies one field name and another summary command specifies a second field name:

- In a report without summary prefix operators, both columns are populated on both summary lines.
- In a report with summary prefix operators, only the specified column is populated on each summary line.

If a prefix operator is used in any summary command, prefix operator processing is required for the request. In most requests it is clear which type of processing to use even if prefix operators are specified in some summary commands but not in others.

Summary Prefix Operators

However, if the first time a summary prefix operator is encountered occurs after a field name has been specified in a summary command *without* an accompanying prefix operator, neither type of processing can be implemented. In this case, by default, processing stops and the following error message is generated:

```
(FOC36376) CANNOT COMBINE SUBTOTAL/RECOMPUTE STYLES WHEN SUMMARYLINES=OLD
```

For example:

```
ON RATING SUBTOTAL COPIES AVE. LISTPR
```

or

```
ON RATING SUBTOTAL LISTPR  
ON CATEGORY SUBTOTAL AVE. COPIES
```

You can eliminate this problem by issuing the SET SUMMARYLINES=NEW command to specify that prefix operator processing should be used. The SUM. operator will then be applied to any field that does have a prefix operator.

Syntax

How to Control Summary Line Processing

```
SET SUMMARYLINES = {OLD|NEW}
```

where:

OLD

Does not allow mixing of summary fields with and without prefix operators when the first field name used in a summary command does not have an associated prefix operator. OLD is the default value. Mixing summary fields with and without prefix operators under these conditions generates the following message:

```
(FOC36376) CANNOT COMBINE SUBTOTAL/RECOMPUTE STYLES WHEN  
SUMMARYLINES=OLD
```

SUBTOTALs (SUB-TOTAL/RECOMPUTE/SUMMARIZE) with and without prefix operators cannot be combined. To allow the mixture - use SET SUMMARYLINES = NEW.

You can specify the SUM. operator for fields for which you want a standard subtotal; this produces the same value that would have been generated without prefix operators.

NEW

Allows mixing of summary fields with and without prefix operators even when the first field name specified in a summary command does not have a prefix operator. All summary fields without prefix operators are processed as though they were specified with the SUM. operator.

Reference Usage Notes for SET SUMMARYLINES

- SET SUMMARYLINES is not supported in a TABLE request (ON TABLE).
- Even if prefix operators are not used on summary lines, report output generated by the two settings for the same request may be slightly different. With SUMMARYLINES NEW, a summary command with a list of field names will populate only those columns on the associated summary line, while SUMMARYLINES OLD will populate every column specified in any summary command.

For example:

```
TABLE FILE MOVIES
PRINT COPIES LISTPR WHOLESALEPR
  BY RATING
  BY CATEGORY
  WHERE CATEGORY EQ 'CHILDREN'
  WHERE RATING EQ 'G'
  ON RATING SUBTOTAL LISTPR AS '*LIST'
  ON CATEGORY SUBTOTAL COPIES AS '*COPY'
END
```

The output when SUMMARYLINES=OLD has subtotals for both COPIES and LISTPR on both sort breaks. WHOLESALEPR is not referenced in either SUBTOTAL command and, therefore, is not on any summary line:

RATING	CATEGORY	COPIES	LISTPR	WHOLESALEPR
G	CHILDREN	2	44.95	29.99
		2	29.95	12.50
		3	26.99	12.00
*COPY	CHILDREN	7	101.89	
*LIST	G	7	101.89	
TOTAL		7	101.89	

Summary Prefix Operators

The output when SUMMARYLINES=NEW has subtotals for COPIES on the CATEGORY sort break and for LISTPR on the RATING sort break. Both columns are populated on the grand total line. WHOLESALEPR is not referenced in either SUBTOTAL command and, therefore, is not on any summary line:

RATING	CATEGORY	COPIES	LISTPR	WHOLESALEPR
G	CHILDREN	2	44.95	29.99
		2	29.95	12.50
		3	26.99	12.00
	*COPY CHILDREN	7		
	*LIST G		101.89	
	TOTAL	7	101.89	

Using Prefix Operators With Calculated Values

If a request includes the RECOMPUTE or SUMMARIZE command, the expression specified in the associated COMPUTE command is applied using values from the summary line. The columns used to recompute the expression can have prefix operators. The recomputed column, regardless of the prefix operator specified for it, applies these input values to the expression specified in the COMPUTE command. Therefore, any supported prefix operator can be specified for the recomputed report column without affecting the calculated result.

With prefix operator processing, all fields used in the COMPUTE command must be calculated by the RECOMPUTE or SUMMARIZE command in order to be populated. If any field used in the expression is not populated, the calculated value returned for the expression is unpredictable.

Example Using Prefix Operators With RECOMPUTE

The first request creates a calculated field named DIFF, which is the difference between DOLLARS and BUDDOLLARS. This value is then recomputed for each region, without using prefix operators:

```
TABLE FILE GGSALLES
SUM UNITS DOLLARS BUDDOLLARS
COMPUTE DIFF/I10 = DOLLARS-BUDDOLLARS;
BY REGION
BY CATEGORY
WHERE CATEGORY EQ 'Food' OR 'Coffee'
WHERE REGION EQ 'West' OR 'Midwest'
ON REGION RECOMPUTE
END
```

The recomputed value is the difference between the totals for DOLLARS and BUDDOLLARS:

Region	Category	Unit Sales	Dollar Sales	Budget Dollars	DIFF
-----	-----	-----	-----	-----	-----
Midwest	Coffee	332777	4178513	4086032	92481
	Food	341414	4338271	4220721	117550
*TOTAL Midwest		674191	8516784	8306753	210031
West	Coffee	356763	4473527	4523963	-50436
	Food	340234	4202338	4183244	19094
*TOTAL West		696997	8675865	8707207	-31342
TOTAL		1371188	17192649	17013960	178689

The following request uses prefix operators in the RECOMPUTE command to calculate maximum DOLLARS and the minimum BUDDOLLARS and then recompute DIFF. No matter which prefix operator we specify for DIFF, it is calculated as the difference between the values in the DOLLARS and BUDDOLLARS columns. If any of the fields used in the calculation (DOLLARS, BUDDOLLARS, and DIFF) do not display on the summary row, the calculation cannot be performed:

```
TABLE FILE GGSales
SUM UNITS DOLLARS BUDDOLLARS
COMPUTE DIFF/I10 = DOLLARS-BUDDOLLARS;
BY REGION
BY CATEGORY
WHERE CATEGORY EQ 'Food' OR 'Coffee'
WHERE REGION EQ 'West' OR 'Midwest'
ON REGION RECOMPUTE MAX. DOLLARS MIN. BUDDOLLARS AVE. DIFF
END
```

Summary Prefix Operators

The output is:

Region	Category	Unit Sales	Dollar Sales	Budget Dollars	DIFF
-----	-----	-----	-----	-----	-----
Midwest	Coffee	332777	4178513	4086032	92481
	Food	341414	4338271	4220721	117550
*TOTAL Midwest			4338271	4086032	252239
West	Coffee	356763	4473527	4523963	-50436
	Food	340234	4202338	4183244	19094
*TOTAL West			4473527	4183244	290283
TOTAL			4473527	4086032	387495

Example Using RECOMPUTE at the Sort Break and Grand Total Levels

The following example adds the ON TABLE RECOMPUTE command to the previous request to calculate the average values for each column. Notice that the value of DIFF is calculated as the difference between the values in the Dollar Sales and the Budget Dollars columns on the grand total line:

```
TABLE FILE GGSALES
SUM UNITS DOLLARS BUDDOLLARS
COMPUTE DIFF/I10 = DOLLARS-BUDDOLLARS;
BY REGION
BY CATEGORY
WHERE CATEGORY EQ 'Food' OR 'Coffee'
WHERE REGION EQ 'West' OR 'Midwest'
ON REGION RECOMPUTE MAX. DOLLARS MIN. BUDDOLLARS DIFF
ON TABLE RECOMPUTE AVE.
END
```

The output is:

Region	Category	Unit Sales	Dollar Sales	Budget Dollars	DIFF
-----	-----	-----	-----	-----	----
Midwest	Coffee	332777	4178513	4086032	92481
	Food	341414	4338271	4220721	117550
*TOTAL Midwest			4338271	4086032	252239
West	Coffee	356763	4473527	4523963	-50436
	Food	340234	4202338	4183244	19094
*TOTAL West			4473527	4183244	290283
TOTAL		342797	4298162	4253490	44672

Using Multiple SUB-TOTAL or SUMMARIZE Commands

SUB-TOTAL and SUMMARIZE propagate their operations to all higher-level sort fields. If a request uses SUB-TOTAL or SUMMARIZE at multiple sort levels, more than one prefix operator may apply to the same field.

When a SUB-TOTAL or SUMMARIZE command on a lower-level sort field propagates up to the higher levels, it applies its prefix operators only to those fields that did not already have a different prefix operator specified at the higher level. For any field that had a prefix operator specified at a higher level, the original prefix operator is applied at the level at which it was first specified and on the grand total line, unless a different operator is specified for the grand total line.

Example Using Multiple SUB-TOTAL Commands

```

DEFINE FILE GGSales
YEAR/YY = DATE;
END
TABLE FILE GGSales
SUM UNITS DOLLARS/D10.2 BUDDOLLARS
BY YEAR
BY ST
BY REGION
BY CATEGORY
WHERE REGION EQ 'West' OR 'Midwest'
WHERE ST EQ 'CA' OR 'IL'
WHERE YEAR EQ '1996' OR '1997'
ON YEAR SUB-TOTAL CNT. UNITS AS '*CNT. UNITS:'
ON ST SUB-TOTAL AVE. DOLLARS AS '*AVE. $:'
ON REGION SUB-TOTAL MIN. AS '*MIN.:'
END

```

Summary Prefix Operators

On the following report output, some of the values have been manually italicized or bolded for clarity:

- The lines preceded by asterisks (*) and the very last (grand total) line are the lines generated by the SUB-TOTAL commands.
- Subtotal values in the normal typeface are the count of unit sales generated by the command ON YEAR SUB-TOTAL CNT. UNITS. This command is the topmost summary command and therefore does not propagate to any other summary lines.
- Subtotal values in bold italic are average dollar sales generated by the command ON ST SUB-TOTAL AVE. DOLLARS. This command is the second summary command and therefore propagates to the DOLLARS column of summary lines for the YEAR sort field.

- Subtotal values in boldface are minimums within their sort groups generated by the command ON REGION SUB-TOTAL MIN. This command is the last summary command and therefore propagates to all other summary lines, but only calculates minimum values for those columns not already populated with a count or an average.

YEAR	State	Region	Category	Unit Sales	Dollar Sales	Budget Dollars
1996	CA	West	Coffee	117539	1,484,873.00	1453548
			Food	116389	1,443,083.00	1414902
			Gifts	74948	947,783.00	946382
			*MIN.: West	74948	947,783.00	946382
			*AVE. \$: CA	74948	1,291,913.00	946382
	IL	Midwest	Coffee	52348	683,559.00	628333
			Food	55867	729,658.00	705915
			Gifts	38405	481,515.00	487090
			*MIN.: Midwest	38405	481,515.00	487090
			*AVE. \$: IL	38405	631,577.33	487090
			*CNT. UNITS: 1996	6	961,745.17	487090
1997	CA	West	Coffee	118044	1,453,023.00	1507092
			Food	106322	1,325,430.00	1302582
			Gifts	77328	988,080.00	961841
			*MIN.: West	77328	988,080.00	961841
			*AVE. \$: CA	77328	1,255,511.00	961841
	IL	Midwest	Coffee	57233	715,220.00	737931
			Food	59293	754,132.00	737912
			Gifts	37918	473,792.00	484611
			*MIN.: Midwest	37918	473,792.00	484611
			*AVE. \$: IL	37918	647,714.67	484611
			*CNT. UNITS: 1997	6	951,612.83	484611
TOTAL				12	956,679.00	484611

TABLASTPAGE

TABLASTPAGE is a system variable that contains the total number of pages in a report. It can be used in conjunction with the page number system variable, TABPAGENO, in a heading, footing, subhead, or subfoot to print "PAGE X of Y" on a PDF, Excel2K, Postscript, or HTML report.

Syntax **How to Display "Page X of Y" in a Report**

```
heading_or_footing_phrase
"Page <TABPAGENO of <TABLASTPAGE"
.
.
.
ON TABLE HOLD AS name FORMAT fmt
```

where:

heading_or_footing_phrase
Can be one of the following:

```
HEADING [CENTER]
FOOTING [CENTER]
ON sortfield SUBHEAD
ON sortfield SUBFOOT
```

sortfield

Is the sort field that triggers a new subheading or subfooting.

name

Is the name of the generated PDF, Excel 2K, postscript, or HTML file.

fmt

Can be one of the following:

```
PDF
EXL2K
PS
POSTSCRIPT
HTML
```

Reference Usage Notes for TABLASTPAGE

- TABLASTPAGE, like TABPAGENO, is a legitimate field that can be used in a COMPUTE. However TABLASTPAGE cannot be used in a DEFINE. DEFINES are evaluated at retrieval time, which is before the number of pages of output is known.
- TABLASTPAGE does not adjust for changes in FOCFIRSTPAGE or for the REPAGE command. For example, if report has 10 pages and the user uses FOCFIRSTPAGE to set the first page number to 3 rather than 1, the value of TABLASTPAGE will still be 10.
- TABLASTPAGE is supported only for a single report, not pooled or compound reports. A separate page count is generated for each report in a compound report.
- TABLASTPAGE is supported only for styled report formats such as HTML, PDF, PS, and EXL2K. It is not supported for WP, DOC, or HTML with STYLE=OFF and STYLEMODE=FIXED.
- TABLASTPAGE causes a second pass through the report results, first to calculate the last page then to print it with TABPAGENO (even when SQUEEZE=OFF).
- TABLASTPAGE does not support the system (external) sort.
- GRAPH FILE does not support TABLASTPAGE.
- TABLEF is not supported with TABLASTPAGE.

Example **Creating an HTML Report That Displays Page X of Y**

```
TABLE FILE GGSALES
"Page <TABPAGE NO of <TABLASTPAGE"
" "
SUM UNITS AS 'Unit,Sales'
BY CATEGORY BY REGION BY PRODUCT
ON CATEGORY PAGE-BREAK
ON TABLE HOLD FORMAT HTML
ON TABLE SET STYLE *
type=report, grid=off , $
type=report, font=helvetica, $
type=heading, style=italic , $
END
```

The output is:

Page 1 of 3

Category	Region	Product	Unit Sales
Coffee	Midwest	Espresso	101154
		Latte	231623
	Northeast	Capuccino	44785
		Espresso	68127
		Latte	222866
	Southeast	Capuccino	73264
		Espresso	68030
		Latte	209654
	West	Capuccino	71168
		Espresso	71675
		Latte	213920

Page 2 of 3

Category	Region	Product	Unit Sales
Food	Midwest	Biscotti	86105
		Croissant	139182
		Scone	116127
	Northeast	Biscotti	145242
		Croissant	137394
		Scone	70732
	Southeast	Biscotti	119594
		Croissant	156456
		Scone	73779
	West	Biscotti	70436
		Croissant	197022
		Scone	72776

Page 3 of 3

Category	Region	Product	Unit Sales
Gifts	Midwest	Coffee Grinder	50393
		Coffee Pot	47156
		Mug	86718
	Northeast	Thermos	46587
		Coffee Grinder	40977
		Coffee Pot	46185
		Mug	91497
	Southeast	Thermos	48870
		Coffee Grinder	47083
		Coffee Pot	49922
		Mug	88474
	West	Thermos	48976
		Coffee Grinder	48081
		Coffee Pot	47432
		Mug	93881
		Thermos	45648

USERFCHK Setting

The USERFCHK setting controls the level of verification applied to DEFINE FUNCTION and Information-Builders-supplied function arguments. It does not affect verification of the number of parameters; the correct number must always be supplied.

Note: This feature has Candidate for Release status in iWay 5.2.0.

Functions typically expect parameters to be a specific type or have a length that depends on the value of another parameter. It is possible in some situations to enforce these rules by converting the type or truncating the length of a parameter and, therefore, avoid generating an error at run-time.

The level of verification and conversion performed depends on the specific function. The following two situations can usually be fixed satisfactorily:

- If a numeric parameter specifies a maximum size for an alphanumeric parameter, but the alphanumeric string supplied is longer than the specified size, the string can be truncated.
- If a parameter supplied as a numeric literal specifies a value larger than the maximum size for a parameter, it can be reduced to the proper size.

Syntax **How to Enable Function Parameter Verification**

Parameter verification can be enabled only for functions supplied by Information Builders. If your site has a locally written function with the same name as an Information-Builder-supplied function, the USERFNS setting determines which function will be used:

```
SET USERFNS= {SYSTEM|LOCAL}
```

where:

SYSTEM

Gives precedence to functions supplied by Information Builders. SYSTEM is the default setting. This setting is also required in order to enable parameter verification.

LOCAL

Gives precedence to locally written functions. Parameter verification is not performed this setting in effect.

Syntax **How to Control Function Parameter Verification**

Issue the following command in FOCPARM, FOC PROF, EDAS PROF, on the command line, in a FOCEXEC, or in an ON TABLE command. Note that the USERFNS=SYSTEM setting must be in effect

```
SET USERFCHK = setting
```

where:

setting

Can be one of the following:

ON

Is the default value. This setting verifies parameters in requests, but does not verify parameters for functions in Master File DEFINES. If a parameter has an incorrect length or type and an attempt is made to fix the problem behind the scenes, the problem is first reported with a warning message. If such a problem cannot be fixed, an error message is generated and the evaluation of the affected expression is terminated.

Because parameters are not verified for functions in a Master File, no errors are reported for those functions until the DEFINE field is used in a subsequent request when, if a problem occurs, the following message is generated:

```
(FOC003) THE FIELDNAME IS NOT RECOGNIZED
```

OFF

Does not verify parameters except in the following cases:

- If a parameter that is too long would overwrite the memory area in which the computational code is stored, the size is automatically reduced without issuing a message.
- If an alphanumeric parameter is too short, it is padded with blanks to the correct length.

FULL

Is the same as ON, but also verifies parameters for functions used in Master File DEFINES.

ALERT

Verifies parameters in a request without halting execution when a problem is detected. It does not verify parameters for functions in Master File DEFINES. If a parameter has an incorrect length or type and an attempt is made to fix the problem behind the scenes, the problem is reported with a warning message. Execution then continues as though the setting were OFF.

Example Verifying Parameters With Correctable Errors

The following request uses SUBSTR to extract the substring that starts in position 6 and ends in position 14 of the TITLE field. The third fifth argument specifies a substring length (500) that is too long (it should be 9):

```
SET USERFCHK = ON
TABLE FILE MOVIES
PRINT TITLE
COMPUTE
  NEWTITLE/A9 = SUBSTR(39, TITLE, 6 ,14, 500, NEWTITLE);
WHERE CATEGORY EQ 'CHILDREN'
END
```

- When the request is executed with USERFCHK=ON, a warning message is produced. The incorrect length is corrected and the request completes processing:

```
(FOC36335) PARAMETER LENGTH CONFLICT IN FUNCTION "SUBSTR", ARG 5.
TITLE                                     NEWTITLE
-----                                     -
SMURFS, THE                               S, THE
SHAGGY DOG, THE                           Y DOG, TH
SCOOBY-DOO-A DOG IN THE RUFF              Y-DOO-A D
ALICE IN WONDERLAND                        IN WONDE
SESAME STREET-BEDTIME STORIES AND SONGS   E STREET-
ROMPER ROOM-ASK MISS MOLLY                R ROOM-AS
SLEEPING BEAUTY                           ING BEAUT
BAMBI
```

- When the request is executed with USERFCHK=OFF, no warning message is produced. The length error is still corrected and processing continues.

Example Verifying Parameters With Uncorrectable Errors

The following request has an incorrect data type in the last argument to SUBSTR. This parameter should specify an alphanumeric field or format for the extracted substring:

```
SET USERFCHK = ON
TABLE FILE MOVIES
PRINT TITLE
COMPUTE
  NEWTITLE/F9 = SUBSTR(39, TITLE, 6 ,14, 500, 'F9');
WHERE CATEGORY EQ 'CHILDREN'
END
```

- When the request is executed with USERFCHK=ON, a message is produced and the request terminates:

```
ERROR AT OR NEAR LINE 5 IN PROCEDURE USERFC3 FOCEXEC
(FOC279) NUMERIC ARGUMENTS IN PLACE WHERE ALPHA ARE CALLED FOR
(FOC009) INCOMPLETE REQUEST STATEMENT
UNKNOWN FOCUS COMMAND WHERE
  BYPASSING TO END OF COMMAND
```

- When the request is executed with USERFCHK=OFF, no verification is done and no message is produced. The request executes and produces incorrect results. In some environments, this type of error may cause abnormal termination of the application:

TITLE	NEWTITLE
-----	-----
SMURFS, THE	*****
SHAGGY DOG, THE	*****
SCOOBY-DOO-A DOG IN THE RUFF	*****
ALICE IN WONDERLAND	1
SESAME STREET-BEDTIME STORIES AND SONGS	-265774
ROMPER ROOM-ASK MISS MOLLY	*****
SLEEPING BEAUTY	*****
BAMBI	0

Wide FOCEXEC Lines

FOCEXECs on MVS can now be stored as members in a PDS with variable length records and a logical record length of up to 4096 bytes. These FOCEXECs are referred to as wide FOCEXECs.

The following are the recommended DCB attributes for a wide FOCEXEC PDS:

- Record Format (RECFM) VB.
- Logical Record Length (LRECL) up to 4096.
- System defined DASD default block size.

Note: This feature has Candidate for Release status in iWay 5.2.0.

Example Using a Wide FOCEXEC Line

In this example, a long qualified field name is used in the print request. The field name has a length of 93:

```
TABLE FILE ADVANT_GARDE_CLOTHING_INTERNATIONAL_SALES_DATASOURCE
PRINT
ADVANT_GARDE_CLOTHING_INTERNATIONAL_SALES_DATASOURCE.FRANCE.PROVINCE_NICE
_STORE500_TOTALSALLES
ADVANT_GARDE_CLOTHING_INTERNATIONAL_SALES_DATASOURCE.FRANCE.PROVINCE_NICE
_STORE501_TOTALSALLES
END
```

The FOCEXEC PDS in which this procedure is a member can be created with an allocation similar to the following:

```
//FOCEXEC DD WIDE.FOCEXEC.DATA,DCB=(RECFM=VB,LRECL=93,BLKSIZE=9300),
//          DSORG=PO,DISP=(NEW,CATLG,DELETE),
//          UNIT=3390,SPACE=(CYL,(20,5,20))
```

Reference Usage Notes for Wide FOCEXEC Lines

- You can allocate, create, and use wide FOCEXECs with the same techniques that apply to standard FOCEXECs. Only the DCB parameters are different.
- Because of the nature of Mainframe retrieval of variable-length records, certain applications may be subject to more overhead with wide FOCEXECs than they would be with fixed length FOCEXECs. To minimize this effect, it is recommended that you allocate PDS libraries containing wide FOCEXECs with large block sizes in order to improve retrieval speed.
- Concatenate PDS libraries containing wide FOCEXECs in front of PDS libraries containing standard FOCEXECs in your CLIST, JCL, or DYNAM ALLOC commands in order to enable cross-platform deployment of applications. If a FOCEXEC from another platform is copied to MVS, MVS will attempt to write the new FOCEXEC to the first data set in the concatenation, which must be a wide FOCEXEC library to accommodate FOCEXECs created on other platforms.
- Sequentially executed wide FOCEXECs may run faster than wide FOCEXECs with -REPEAT loops.
- Caution should be given to syntax in which blanks at the end of a line are meaningful for continuation lines, such as in -SET of Dialogue Manager variables and HEADING lines.
- If you try to access a wide FOCEXEC PDS in a release prior to iWay 5.2.0, an 0C4 abend will result.

Reference Messages for Wide FOCEXEC Lines

The following error message will be issued when the LRECL of a FOCEXEC library exceeds 4096:

```
(FOCxxx)    FOCEXEC LIBRARY EXCEEDS THE LRECL LIMIT: Focexec library name
            THE MAXIMUM RECORD LENGTH OF A FOCEXEC LIBRARY CANNOT EXCEED
            4096
```

The following error message will be issued when a PDS cannot support a wide FOCEXEC:

```
(FOCxxx)    PDS LIBRARY CANNOT SUPPORT FOCEXEC: Focexec name
            A PDS LIBRARY MUST BE ALLOCATED WITH A VARIABLE BLOCK FORMAT WITH
            AN LRECL OF UP TO 4096 IN ORDER TO USE A WIDE FOCEXEC
```

CHAPTER 11

SAP/IDoc Processing

Topics:

- IDoc Listener

This section covers new features for SAP/IDOC Processing.

IDoc Listener

Intermediate Documents (IDocs) and the IDoc Listener provide a means to perform Electronic Data Interchange (EDI) with SAP. EDI-enabled applications in SAP are capable of generating IDoc data from an SAP document. IDocs can be sent either in batch or real time, and are perfect for online data warehousing.

The purpose of the IDoc Listener is to generate a Relational Data Base Management System (RDBMS) environment that replicates the source SAP RDBMS tables from which the IDoc originates. Those tables are then updated automatically as new IDocs are received.

For more information on configuration and operation, see the IDoc Listener documentation in the *iWay Server Administration Guide*.

CHAPTER 12

XFocus

Topics:

- XFocus Overview
- Using XML FOCUS to Execute XML Documents
- Using XML FOCUS to Archive XML Documents

XFocus adds new functionality that expands your capabilities by allowing you to receive, archive and process XML files. It also allows you to send an executable request contained in an XML document.

XFocus Overview

The new XFocus feature provides the server with the ability to receive XML documents via an MQseries queue or from a SOAP protocol stream and then be able to perform one of two operations, EXECUTE or ARCHIVE. The XFocus feature is configurable through the Web console. This feature is available on UNIX, OS/390 and z/OS, NT/2000, OS/400 and Open/VMS.

Using XML FOCUS to Execute XML Documents

The EXECUTE operation takes an XML document, extracts a request (Data retrieval or RPC), processes the request and returns an answer set in XML format to the calling environment. The original XML document must conform to the eda.dtd.

Example Using XFocus to Execute an Operation in an XML Document

The following is an example of how an EXECUTE operation is used against an XML document to process a SELECT statement.

```
<?xml version="1.0" ?>
<!DOCTYPE xmlfoc01 SYSTEM "eda.dtd">
  <eda>
    <request>
      <connection>
        <dsn>unused</dsn>
        <user>unused</user>
        <password>unused</password>
        <sql>
          <query>select car from car</query>
        </sql>
      </connection>
    </request>
  </eda>
```

Using XML FOCUS to Archive XML Documents

The ARCHIVE operation takes any XML document, extracts its content and, preserving the hierarchical structure, stores it in a FOCUS database. The database can then be queried from a Business Intelligence viewpoint or the XML can be extracted in its original format for further processing. In order to archive XML documents the server must be configured to enable this feature.

Procedure Using XML FOCUS to Archive XML Documents

The following is an overview of the steps necessary to archive XML documents using XFocus.

1. A listener must be enabled. This feature allows you to pick one or both of the listeners available MQseries or a SOAP listeners.
2. A service must be configured for the dtd. This one to one relationship is done for performance. You have to configure one service for each dtd you will archive.
3. A synonym must be created against the source dtd of the XML document. A load procedure and target file for archiving will be created.

Note: Steps 2 and 3 will need to be performed for each unique XML (dtd document) to be archived.

Index

A

Access Beans Support 1-11
active connections 5-2
ADABAS Data Adapter 1-2
 format buffers and 1-3
 Internal Sequence Number (ISN) and 1-3
 ORDER fields in 1-2
 Super Descriptors and 1-2
agent services 5-2
ALLBASE Data Adapter 1-4
alpha fields 1-18
 restricting size 1-18

B

B APIs (Business Application Programming Interfaces) 1-14
Bulk Insert API 1-5
bulk load capacity 7-4
BULL GCOS Data Adapter 1-18
 accessing files with 1-18
 enhancements 1-19
BULL GCOS TDS/TP8 Transaction Adapter 1-19
Business Application Programming Interfaces (B APIs) 1-14
BW Reporting 1-15

C

CA certificates 6-3
CDN command 10-81
CHAR2INT function 8-4
CLOB fields 1-16

COBOL Copy Book 1-5
column width 10-18
COM data source 10-84 to 10-85
COMBINE command 10-2 to 10-3
 structure 10-3
COMT data source 10-84 to 10-85
connections 5-2
 active 5-2
 queued 5-2
CONTAINS command 8-2
 computer fields and 8-2
CONTAINS test 1-9
continental decimal notation 10-81
CREATE SYNONYM command 1-4, 1-8, 2-2
 restrictions 1-18
CREATE SYNONYM command and 1-18
Create Synonym Support 1-11
currency symbol 10-32
 format options 10-33
 support 10-32

D

data space 10-26
Database Management Systems (DBMSs) 1-1
DATASET attribute 10-10 to 10-12
 messages 10-12
DB2 CLOB columns 1-9
 adding 1-9
 describing 1-9
DB2 Data Adapter 1-16 to 1-17
 table types on OS/400 1-17

Index

DBFILE Data Adapter 1-16
 and COBOL 1-16
 and RPG 1-16

DBMS return codes 8-3

DBMSs (Database Management Systems) 1-1

decimal alignment 10-13
 footings and 10-17
 headings and 10-13, 10-17

DEFINE command 10-4 to 10-6
 compiling 10-5
 COMPUTE commands and 10-5

Direct SQL Passthru 1-18

directory tree structure 5-5

DYNAM command 2-3

DYNAM commands 2-4

DYNAM DSIO command
 messages 10-30

DYNAM DSIO ENABLE command 10-29

dynamic data space 10-19 to 10-20, 10-26

E

edastart -console option 5-4

-enable newconnect 5-4

ENGINE function 8-3

Enterprise Java Beans 1-11

Entity Java Beans 1-11

ERROROUT command 10-31

ESSBASE Data Adapter 1-7

ETL Manager 7-1

ETL requests 7-2
 logs 7-2
 managing from the Web Console 7-3
 migrating from a server 7-4
 migrating from Workbench 7-4
 saving as procedures 7-2
 scheduling 7-3
 synonyms 7-2

ETL scheduler 7-3
 managing from the Web Console 7-3

exit on error 10-31

F

FAILURE terminal 3-3

FETCH FIRST n ROWS clause 1-6 to 1-7

field names 10-62
 long 10-62

FIELDTYPE=R attribute 10-35

FML hierarchies 10-39
 error messages 10-52

FOCBS 10-53

FOCFIRTSPAGE command 10-6 to 10-7

FOCUS Database Server
 MVS and 10-53

FOLD-LINE attribute 10-68
 multiple instances 10-68

FOR statement 10-70
 repeated values 10-70

G

Global Format Buffer ID (GFBID) 1-3

H

Hiper Memory 2-4
 HNODATA command 10-57 to 10-58
 HOLDMISS command 10-61
 HTTP Basic Authentication 6-3
 HUB/SUB servers 1-2

I

IBI Subsystem 10-53
 IDoc listener 11-2
 Informix data type 1-10
 INSERTSIZE value 1-5
 INT2CHAR function 8-4
 Intermediate Documents (IDocs) 11-2
 Internal Sequence Number (ISN) 1-3

J

job ID 7-3
 joins 1-15
 between data sources 1-15
 between DBMSs 1-15

K

keywords 5-2
 max_connections_per_user 5-2
 profile 5-2

L

Last Command column 8-2
 listener statistics 5-5
 Log tables 7-3
 long qualified field names 10-62
 LVARCHAR columns 1-10

iWay Summary of New Features

M

Maintain 8-1 to 8-2
 DBMS return codes 8-3
 file extensions 10-38
 functions 8-4
 Language and Server environment 8-1
 system sets 8-2
 MAINTAIN FILETYPE extension 8-4
 Master File access 10-35
 read-only 10-35
 max_connections_per_user keyword 5-2
 metadata generation 1-5
 options for 1-5
 Metadata Manager 7-4
 Micro Focus C-ISAM Data Adapter 1-11
 minus edit format option 10-65, 10-67
 missing data characters 10-57 to 10-59
 MNT file extension 10-38
 MQSI XML SupportPac 3-2
 installing 3-3
 usage notes 3-2
 MQXML listener 3-2
 configuring 3-2
 multiple flat files 7-3
 multiple target tables 7-3
 multi-variate regress 10-72 to 10-73
 MUMPS Data Adapter 1-12
 MVS External Sort 2-3
 MVSAPP feature 2-4

N

Native Interface support 1-14
NEXT command 8-2
NLSCHR function 8-4
non-Focus files 4-2
NORMDST function 10-76
NORMSINV function 10-76

O

ODBC connection 1-12
OMITS test 1-9
OMITS value 8-2
OpenVMS Operating System 4-2
ORDER fields 1-2
OS/400 Operating System 4-2
OUTPUT terminal 3-3
output terminals 3-2
 FAILURE 3-3
 OUTPUT 3-3
 RETRY 3-3

P

page numbering 10-6
platform support 4-1
plug-in nodes 3-2
 installing 3-3
 output terminals 3-3
 usage notes 3-4
prefix operators 10-86, 10-88, 10-94
profile keyword 5-2
profiles directory 5-5
Progress 9.1 Data Adapter 1-12

Q

queued connections 5-2
-quiesce newconnect 5-4
-quit option 5-4

R

READLIMIT tests 1-6
read-only access 10-35
RECOMPUTE command 10-86, 10-88, 10-94
REGRESS method 10-72
Remote Procedure Calls (RPCs) 2-3 to 2-4
request logs 7-2
Resource Analyzer 9-2
Resource Governor 9-2
RESTRICTSUM command 1-7
RETRY terminal 3-3

S

SAP BW Data Adapter 1-15
SAP processing 11-2
SAP R/3 Interface 1-13
 BAPI support for 1-14
 Native Interface enhancement 1-13
 Native Interface support 1-14
SAVEDMASTERS command 10-82, 10-84

- security 6-1
 - administration levels 6-3
 - administrator privileges 6-3
 - methods 6-2
 - on Windows NT/2000 6-2
 - passwords 6-2
 - specifying Windows NT/2000 clients 6-2
 - SSL (Secure Socket Layer) 6-3
 - TLS (Transport Layer Security) 6-3
 - user privileges 6-3
 - user rights 6-2
 - WCPROTECT mode 6-4
 - security settings 2-5
 - INTERNAL type 2-5
 - MVS type 2-5
 - server configuration file 6-2
 - Server for OS/390 and z/OS 2-1
 - accessing synonyms from 2-2
 - CREATE SYNONYM command 2-2
 - DYNAM command 2-3
 - enhancements 2-4
 - MVS External Sort 2-3
 - security settings 2-5
 - supported data adapters 2-2
 - Server level profiles 5-5
 - servers 5-1
 - configuring 5-4
 - Enable control 5-4
 - managing connections to 5-4
 - parameters for 5-2
 - Quiesce control 5-4
 - user level profiles 5-5
 - SET MEASURE command 1-7
 - SET option 8-2
 - SET SYNONYM command 1-2
 - Shared Library Path 5-5
 - SMF records 2-5
 - SOAP/XML listener 3-4
 - configuring 3-4
 - requirements 3-5
 - SQL SELECT statements 7-5
 - SQL/Answer set object 7-5
 - SSL (Secure Socket Layer) 6-3
 - Statistics tables 7-3
 - SUB-TOTAL command 10-86, 10-88, 10-94
 - SUBTOTAL command 10-86, 10-88, 10-94
 - SUFFIX values 10-84 to 10-85
 - SUMMARIZE command 10-86, 10-88, 10-94
 - Super Descriptors 1-2
 - synonyms 2-2
 - accessing from OS/390 and z/OS 2-2
 - creating through the Server Console 2-2
- T**
-
- TABLSTPAGE variable 10-100 to 10-101
 - table services 10-1
 - TABT data source 10-84 to 10-85
 - Text Console 5-4
 - TLS (Transport Layer Security) 6-3
 - TX fields 1-8 to 1-9, 1-16
 - describing 1-8
 - relational data adapters and 1-8
 - type on EDAPRINT command 8-6

U

UniVerse Data Adapter 1-8

UNIX Operating System 4-2

user level profiles 5-5

user rights 6-2

USERFCHK setting 10-103

USERFNS setting 10-104

V

VARCHAR support 1-16

W

WCPROTECT mode 6-4

Web Console 8-2

WHERE clause 1-8 to 1-9

wide procedure lines 10-107

WITH STATISTICS option 1-15

Workbench 7-5

X

XA support 1-2

 HUB/SUB servers and 1-2

XFocus 12-2

 ARCHIVE operation and 12-3

 EXECUTE operation and 12-2

XMI Client/Server Data Adapter 4-2

 configuring 4-3

XMI listener 4-2 to 4-3

XML documents 12-2

 XFocus and 12-2

XML FOCUS 12-2

Reader Comments

In an ongoing effort to produce effective documentation, the Documentation Services staff at Information Builders welcomes any opinion you can offer regarding this manual.

Please use this form to relay suggestions for improving this publication or to alert us to corrections. Identify specific pages where applicable. You can contact us through the following methods:

Mail: Documentation Services - Customer Support
Information Builders, Inc.
Two Penn Plaza
New York, NY 10121-2898

Fax: (212) 967-0460

E-mail: books_info@ibi.com

Web form: <http://www.informationbuilders.com/bookstore/derf.html>

Name: _____

Company: _____

Address: _____

Telephone: _____ Date: _____

E-mail: _____

Comments:

Reader Comments