TIBCO FOCUS®

Release Notes

Release 8207.27.0
March 2021
DN1001076.0321

Contents

Chapter 1

New Features

The following topic provides information about new features in this release of TIBCO FOCUS®.

In this chapter:

Reporting Enhancements

This topic describes enhancements that can be used for reporting.

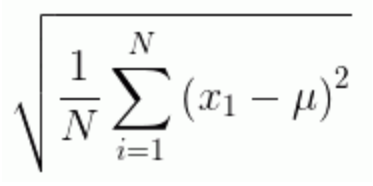
Full Outer Join Support

Full outer joins and right outer joins are now supported whether or not the underlying data source supports them. When the underlying data source has support for these joins, the join processing is passed to the database engine. When it does not support them, all necessary data is returned and the join processing is handled by FOCUS.

Standard Deviation Prefix Operators: STDP. and STDS.

The standard deviation prefix operators return a numeric value that represents the amount of dispersion in the data. The set of data can be specified as the entire population (STDP.) or a sample (STDS.). The standard deviation is the square root of the variance, which is a measure of how observations deviate from their expected value (mean). If specified as a population, the divisor in the standard deviation calculation (also called degrees of freedom) will be the total number of data points, N. If specified as a sample, the divisor will be N-1.

If x_i is an observation, N is the number of observations, and μ is the mean of all of the observations, the formula for calculating the standard deviation for a population is:



$$\sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_1-\mu)^2}$$

To calculate the standard deviation for a sample, the mean is calculated using the sample observations, and the divisor is N-1 instead of N.

Syntax: How to Calculate the Standard Deviation Using Prefix Operators

To calculate the standard deviation for a population, the syntax is:

```
STDP.field
```

To calculate the standard deviation for a sample, the syntax is:

```
STDS.field
```

where:

field
Numeric

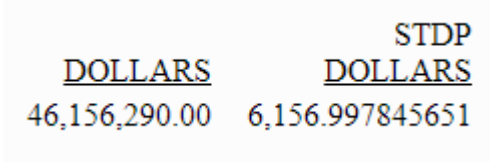
Is the set of observations for the standard deviation calculation.

Example: Calculating the Standard Deviation of a Population

The following request calculates the standard deviation of the population of the DOLLARS field converted to double precision.

```
DEFINE FILE ibisamp/ggsales
DOLLARS/D12.2 = DOLLARS;
END
TABLE FILE ibisamp/ggsales
SUM DOLLARS STDP.DOLLARS
ON TABLE SET PAGE NOLEAD
ON TABLE SET STYLE *
GRID=OFF,$
ENDSTYLE
END
```

The output is shown in the following image.



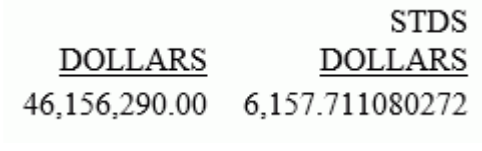
DOLLARS	STDP DOLLARS
46,156,290.00	6,156.997845651

Example: **Calculating the Standard Deviation of a Sample**

The following request calculates the standard deviation of a sample of the DOLLARS field converted to double precision.

```
DEFINE FILE ibisamp/ggsales
DOLLARS/D12.2 = DOLLARS;
END
TABLE FILE ibisamp/ggsales
SUM DOLLARS STDS.DOLLARS
ON TABLE SET PAGE NOLEAD
ON TABLE SET STYLE *
GRID=OFF,$
ENDSTYLE
END
```

The output is shown in the following image.



DOLLARS	STDS DOLLARS
46,156,290.00	6,157.711080272

Using a WHERE Phrase in a Filter on a Calculated Value

In prior releases, specifying a WHERE phrase on a calculated value (COMPUTE field) generated an error, and processing was terminated. In the current release, a WHERE condition on a calculated field will be automatically replaced with WHERE_GROUPED if the filter is eligible for WHERE_GROUPED processing, or with WHERE TOTAL if it is not.

Navigating Joins Between Cluster Synonyms

By default, when joining cluster synonyms, a hierarchy of segments is constructed from all of the joined files, and the resulting hierarchy is navigated in top-to-bottom, left-to-right order.

Therefore, if a left outer join is specified from a host synonym to a cluster that has an inner join, the inner join will be performed last and may remove rows from the host file, counteracting the purpose of the left outer join. Using the SET FOCTRANSFORM = NESTED_CLUSTERS/ON command, you can force the joins in the target cluster to be performed prior to the join between the host and target synonyms. When you use this setting, SQL scripts are used to join the tables in the target cluster prior to implementing the join to the host file. The left outer join will be performed last and will retain all rows in the host synonym.

Note: This feature is only supported in FOCUS for Distributed Systems.

The syntax is:

```
SET FOCTRANSFORM = {NESTED_CLUSTERS/OFF|NESTED_CLUSTERS/ON}
```

where:

NESTED_CLUSTERS/OFF
: Maintains the left-to-right, top-to-bottom order of segment navigation. This is the default value.

NESTED_CLUSTERS/ON
: Performs the joins in the target cluster synonym prior to joining the host synonym to the result.

Reference: Usage Notes for Joins to Cluster Synonyms

- This feature requires that the joins be optimized. The command SET SHORTPATH = SQL must be in effect for combinations of inner and outer joins with the setting FOCTRANSFORM = NESTED_CLUSTERS/OFF, in order for the request to be optimized. The SHORTPATH = SQL setting has no effect on optimization with the setting FOCTRANSFORM = NESTED_CLUSTERS/ON.
- You cannot join to a non-root segment of a cluster synonym. If you issue a join to a non-root segment, the following message displays and the request terminates:

  ```
  (FOC906) JOIN TO NON-ROOT SEGMENT segname IS NOT ALLOWED FOR
  NESTED_CLUSTERS
  ```

Example: Navigating Joins Between Cluster Synonyms

This example uses SQL Server data sources generated from a file of citibike trips uploaded from *https://www.citibikenyc.com/system-data*, and from a file of zip codes for the stations used for the trips (you can download this file from *https://techsupport.informationbuilders.com/public/station_zip.csv*).

A cluster synonym named station_trip_cls joins the station zip data source to a data source containing partial trip data (with only a few rows). The following shows the inner join defined in the cluster synonym:

```
FILENAME=STATION_TRIP_CLS, $
  SEGMENT=STATION_ZIP_OLEDB, CRFILE=CITIBIKE/STATION_ZIP_OLEDB,
CRINCLUDE=ALL, $
  SEGMENT=CITIBIKE_PARTIAL_OLEDB, SEGTYPE=KU, PARENT=STATION_ZIP_OLEDB,
    CRFILE=CITIBIKE/CITIBIKE_PARTIAL_OLEDB, CRINCLUDE=ALL, CRJOINTYPE=INNER,
    JOIN_WHERE=STATION_ID EQ START_STATION_ID;, $
```

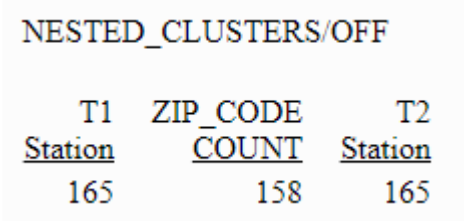
The following request issues a left outer join from a larger version of the trip data file to the cluster:

```
SET FOCTRANSFORM = NESTED_CLUSTERS/&VALUE
SET SHORTPATH = SQL
JOIN LEFT_OUTER START_STATION_ID IN CITIBIKE_TRIPDATA TAG T1 TO ALL
STATION_ID IN STATION_TRIP_CLS TAG T2 AS J1
TABLE FILE CITIBIKE_TRIPDATA
" NESTED_CLUSTERS/&VALUE"
" "
SUM CNT.T1.START_STATION_ID AS T1,Station CNT.ZIP_CODE
CNT.T2.START_STATION_ID AS T2,Station
ON TABLE SET PAGE NOLEAD
ON TABLE SET STYLE *
GRID=OFF,$
ENDSTYLE
END
```

Running the request with &VALUE set to OFF generates the following trace:

```
SELECT
   COUNT(T1."START_STATION_ID"),
   COUNT(T2."ZIP_CODE"),
   COUNT(T3."START_STATION_ID")
   FROM
   ( ( citibike_tripdata_mssqloledb T1
   LEFT OUTER JOIN
   station_zip_oledb T2
   ON T2."STATION_ID" = T1."START_STATION_ID" )
   INNER JOIN
   citibike_partial_msoledb T3
   ON (T3."START_STATION_ID" = T2."STATION_ID") );
```

The output is shown in the following image. The inner join was done last, reducing the number of stations in the host file to the same number as in the cluster.



NESTED_CLUSTERS/OFF

T1 Station	ZIP_CODE COUNT	T2 Station
165	158	165

Running the request with &VALUE set to ON generates the following trace. Two SQL scripts are generated, one for the host file and one for the join in the cluster. Then, the left outer join is performed against the result of the inner join:

```
SELECT
   T1."START_STATION_ID" AS "SK001_START_STATION_ID",
    COUNT(T1."START_STATION_ID") AS "VB001_CNT_START_STATION_ID"
    FROM
   citibike_tripdata_mssqloledb T1
    GROUP BY
   T1."START_STATION_ID";
   (FOC2546) SQL SCRIPT
__CITIBIKE_TRIPDATA_OLEDB_CITIBIKE_TRIPDATA_OLEDB.SQL CREATED SUCCESSFULLY
(BUT NOT EXECUTED)
   _EDATEMP/__citibike_tripdata_oledb_citibike_tripdata_oledb HELD AS
SQL_SCRIPT
    SELECT
  T1."STATION_ID" AS "SK001_STATION_ID",
    COUNT(T1."ZIP_CODE") AS "VB001_CNT_ZIP_CODE",
    COUNT(T2."START_STATION_ID") AS "VB002_CNT_START_STATION_ID"
    FROM
   station_zip_oledb T1,
   citibike_partial_msoledb T2
    WHERE
   (T2."START_STATION_ID" = T1."STATION_ID")
    GROUP BY
   T1."STATION_ID";
  (FOC2546) SQL SCRIPT
__CITIBIKE_TRIPDATA_OLEDB_STATION_PARTIAL_OLEDB_CLS.SQL CREATED
SUCCESSFULLY (BUT NOT EXECUTED)
  _EDATEMP/__citibike_tripdata_oledb_station_partial_oledb_cls HELD AS
SQL_SCRIPT
```

```
 SELECT
 SUM(T1."VB001_CNT_START_STATION_ID"),
 SUM(T2."VB001_CNT_ZIP_CODE"),
 SUM(T2."VB002_CNT_START_STATION_ID")
 FROM
(
( /* vvv */
   SELECT
  T1."START_STATION_ID" AS "SK001_START_STATION_ID",
   COUNT(T1."START_STATION_ID") AS
  "VB001_CNT_START_STATION_ID"
   FROM
  citibike_tripdata_mssqloledb T1
   GROUP BY
  T1."START_STATION_ID"
) /* ^^^ */ T1
 LEFT OUTER JOIN
( /* vvv */
   SELECT
  T1."STATION_ID" AS "SK001_STATION_ID",
   COUNT(T1."ZIP_CODE") AS "VB001_CNT_ZIP_CODE",
   COUNT(T2."START_STATION_ID") AS
  "VB002_CNT_START_STATION_ID"
   FROM
  station_zip_oledb T1,
  citibike_partial_msoledb T2
   WHERE
  (T2."START_STATION_ID" = T1."STATION_ID")
   GROUP BY
  T1."STATION_ID"
) /* ^^^ */ T2
 ON T2."SK001_STATION_ID" = T1."SK001_START_STATION_ID" );
```

The output is shown in the following image. The left outer join was done last, maintaining the original number of stations in the host file.

NESTED_CLUSTERS/ON

T1	ZIP_CODE	T2
Station	COUNT	Station
6680	8	9

General Enhancements

This topic describes enhancements that can be used in multiple contexts.

Enhancements to DATE_ORDER

The DATE_ORDER parameter for date-time formats is now supported when the format specifies month translation, zero suppression or removal, or the comma option. For some formats with the comma option, reordering the date components may require elimination of the comma.

For example, the following request creates date fields with zero removal and suppression (YYMDoe), month translation and zero suppression (YYMte), and month translation with a comma and space between the month and year (HMTkYY). The DATE_ORDER is set to DMY:

```
-DEFAULT &ORDER=DMY;
SET DATE_ORDER=&ORDER
DEFINE FILE GGSALES
ORIGINAL/YYMD=20190704;
YYMDoe/HYYoe=DT(2019/07/04);
YYMte/HYYMte=YYMD;
YYMTDk/HMTkYY=YYMD;
END
TABLE FILE GGSALES
SUM ORIGINAL YYMDoe YYMte YYMTDk
BY CATEGORY
ON TABLE SET PAGE NOLEAD
ON TABLE SET STYLE *
GRID=OFF,$
ENDSTYLE
END
```

The output is shown in the following image.

Category	ORIGINAL	YYMDoe	YYMte	YYMTDek
Coffee	04/07/2019	4/7/2019	4 Jul 2019	July, 2019
Food	04/07/2019	4/7/2019	4 Jul 2019	July, 2019
Gifts	04/07/2019	4/7/2019	4 Jul 2019	July, 2019

International System (SI) Numeric Format Abbreviation Options

The International System standard provides numeric abbreviations for very large and very small numbers.

FOCUS supports the following SI-compliant numeric abbreviations. The SI-compliant format uses a two-character display code that consists of a lowercase *n* followed by the SI abbreviation.

Prefix	WebFOCUS Format Code	Size	Example	English Name (American/British)
yotta	nY	10**24	1000000000000000000000000	septillion/quadrillion
zetta	nZ	10**21	1000000000000000000000	sextillion/trilliard
exa	nE	10**18	1000000000000000000	quintillion/trillion
peta	nP	10**15	1000000000000000	quadrillion/billiard
tera	nT	10**12	1000000000000	trillion/billion
giga	nG	10**9	1000000000	billion/milliard
mega	nM	10**6	1000000	million
kilo	nK	10**3	1000	thousand
milli	nm	10**(-3)	0.001	thousandth
micro	nu	10**(-6)	0.000001	millionth
nano	nn	10**(-9)	0.000000001	billionth/milliardth
pico	np	10**(-12)	0.000000000001	trillionth/billionth
femto	nf	10**(-15)	0.000000000000001	quadrillionth/billiardth
atto	na	10**(-18)	0.000000000000000001	quintillionth/trillionth
zepto	nz	10**(-21)	0.000000000000000001	sextillionth/trilliardth
yocto	ny	10**(-24)	0.000000000000000000000001	septillionth/quadrillionth

The following request uses the mega and giga format options. The decimal precision is controlled by the format which, in this case, is a reformat specified in the SUM command.

```
DEFINE FILE GGSALES
NEWDOLL/D12.2 = DOLLARS * 100;
END
TABLE FILE GGSALES
SUM DOLLARS NEWDOLL/D12.5nM AS Millions NEWDOLL/D12.3nG AS Billions
BY CATEGORY
ON TABLE SET PAGE NOLEAD
ON TABLE SET STYLE *
GRID=OFF,$
ENDSTYLE
END
```

The output is shown in the following image.

Category	Dollar Sales	Millions	Billions
Coffee	17231455	1,723.14550M	1.723G
Food	17229333	1,722.93330M	1.723G
Gifts	11695502	1,169.55020M	1.170G

New Regular Expression Pattern Matching Functions

The following FOCUS and SQL functions search for strings that match a pattern expressed as a regular expression.

- **REGEXP_COUNT.** Counts the number of matches to a regular expression pattern in a string.
- **REGEXP_INSTR.** Returns the first position of a regular expression pattern in a string.
- **REGEXP_REPLACE.** Replaces all matches to a regular expression pattern in a string with a replacement string.
- **REGEXP_SUBSTR.** Returns the first match to a regular expression pattern in a string.

You can search online for information about the symbols used to create a regular expression pattern. For example, Wikipedia has a good introduction at:

https://en.wikipedia.org/wiki/Regular_expression

REGEXP_COUNT: Counting the Number of Matches to a Pattern in a String

REGEXP_COUNT returns the integer count of matches to a specified regular expression pattern within a source string.

Syntax: **How to Count the Number of Matches to a Pattern in a String**

```
REGEXP_COUNT(string, pattern)
```

where:

string
Alphanumeric

Is the input string to be searched.

pattern
Alphanumeric

Is the regular expression pattern to match.

Example: **Counting the Number of Matches to a Pattern in a String**

The following examples use the following Regular Expression symbols.

- $, which searches for a specified expression that occurs at the end of a string.
- ^, which searches for a specified expression that occurs at the beginning of a string.

REGEXP_COUNT counts the number of occurrences of the characters 'umpty' that occur at the end of the string 'Humpty Dumpty'.

```
REGEXP_COUNT('Humpty Dumpty', 'umpty$')
```

The result is 1.

REGEXP_COUNT counts the number of occurrences of the characters 'umpty' that occur at the beginning of the string 'Humpty Dumpty'.

```
REGEXP_COUNT('Humpty Dumpty', '^umpty')
```

The result is 0.

REGEXP_INSTR: Returning the First Position of a Pattern in a String

REGEXP_INSTR returns the integer position of the first match to a specified regular expression pattern within a source string. The first character position in a string is indicated by the value 1. If there is no match within the source string, the value 0 is returned.

Syntax: **How to Return the Position of a Pattern in a String**

```
REGEXP_INSTR(string, pattern)
```

where:

string
Alphanumeric

Is the input string to be searched.

pattern
Alphanumeric

Is the regular expression pattern to match.

Example: Finding the Position of a Pattern in a String

The following examples use the following Regular Expression symbols.

- $, which searches for a specified expression that occurs at the end of a string.
- ^, which searches for a specified expression that occurs at the beginning of a string.

REGEXP_INSTR finds the position of the characters 'umpty' that occur at the end of the string 'Humpty Dumpty'.

```
REGEXP_INSTR('Humpty Dumpty', 'umpty$')
```

The result is 9.

REGEXP_INSTR finds the position of the characters 'umpty' that occur at the beginning of the string 'Humpty Dumpty'.

```
REGEXP_INSTR('Humpty Dumpty', '^umpty')
```

The result is 0.

REGEXP_REPLACE: Replacing All Matches to a Pattern in a String

REGEXP_REPLACE returns a string generated by replacing all matches to a regular expression pattern in the source string with the given replacement string. The replacement string can be a null string.

Syntax: How to Replace Matches to a Pattern in a String

```
REGEXP_REPLACE(string, pattern, replacement)
```

where:

string
Alphanumeric

Is the input string to be searched.

pattern
Alphanumeric

Is the regular expression pattern to match.

replacement
Alphanumeric

Is the replacement string.

Example: **Replacing Matches to a Pattern in a String**

The following example uses the following Regular Expression symbol.

- ^, which searches for a specified expression that occurs at the beginning of a string.

REGEXP_REPLACE replaces the characters 'ENG' at the beginning of the field COUNTRY with the replacement string 'SCOT'.

```
REGEXP_REPLACE(COUNTRY, '^ENG', 'SCOT')
```

For 'ENGLAND', the result is 'SCOTLAND'.

REGEXP_SUBSTR: Returning the First Match to a Pattern in a String

REGEXP_SUBSTR returns a string that contains the first match to a specified regular expression pattern within a source string. If there is no match within the source string, a null string is returned.

Syntax: **How to Returning the First Match to a Pattern in a String**

```
REGEXP_SUBSTR(string, pattern)
```

where:

string
Alphanumeric

Is the input string to be searched.

pattern
Alphanumeric

Is the regular expression pattern to match.

Example: Returning the First Match of a Pattern in a String

The following example uses the following Regular Expression symbols.

- [A-Z], which matches any uppercase letter.
- $, which searches for a specified expression that occurs at the end of a string.

REGEXP_SUBSTR searches for a string with any uppercase letter followed by the characters 'umpty' at the end of the string 'Humpty Dumpty'.

```
REGEXP_SUBSTR('Humpty Dumpty', '[A-Z]umpty$')
```

The result is 'Dumpty'.

New Functions for Date-time Conversion Between Local and UTC Time

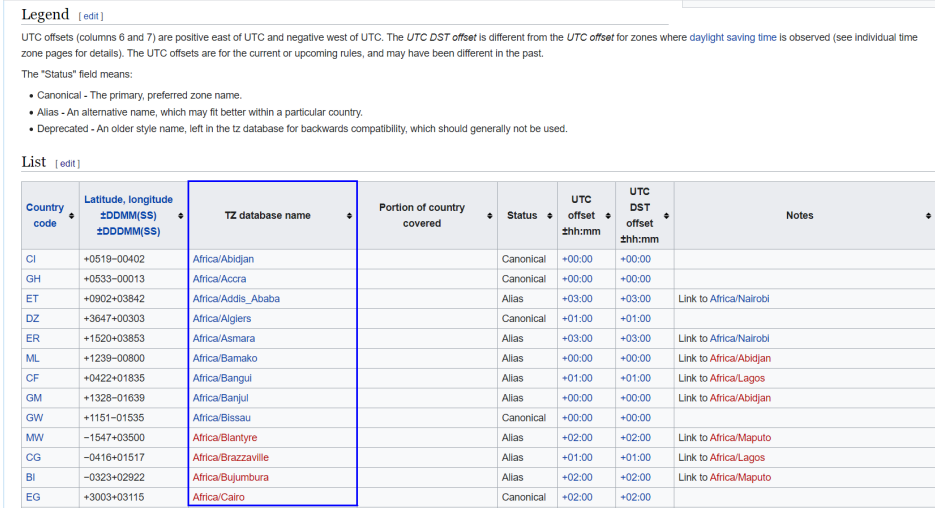
Coordinated Universal Time (UTC) is the time standard commonly used around the world. To convert UTC time to a local time, a certain number of hours must be added to or subtracted from the UTC time, depending on the number of time zones between the locality and Greenwich, England (GMT).

The following functions convert date-time values between UTC time and local time.

- **DT_TOUTC.** Converts local time to UTC time.
- **DT_TOLOCAL.** Converts UTC time to local time.

Converting timestamp values from different localities to a common standard time enables you to sort events into the actual event sequence.

These functions require IANA (Internet Assigned Numbers Authority) time zone database names (expressed as 'Area/Location') as parameters. You can find a table of IANA TZ database names on Wikipedia at *https://en.wikipedia.org/wiki/List_of_tz_database_time_zones*, as shown in the following image.



Legend [edit]

UTC offsets (columns 6 and 7) are positive east of UTC and negative west of UTC. The *UTC DST offset* is different from the *UTC offset* for zones where daylight saving time is observed (see individual time zone pages for details). The UTC offsets are for the current or upcoming rules, and may have been different in the past.

The "Status" field means:

- Canonical - The primary, preferred zone name.
- Alias - An alternative name, which may fit better within a particular country.
- Deprecated - An older style name, left in the tz database for backwards compatibility, which should generally not be used.

List [edit]

Country code	Latitude, longitude ±DDMM(SS) ±DDDMM(SS)	TZ database name	Portion of country covered	Status	UTC offset ±hh:mm	UTC DST offset ±hh:mm	Notes
CI	+0519−00402	Africa/Abidjan		Canonical	+00:00	+00:00	
GH	+0533−00013	Africa/Accra		Canonical	+00:00	+00:00	
ET	+0902+03842	Africa/Addis_Ababa		Alias	+03:00	+03:00	Link to Africa/Nairobi
DZ	+3647+00303	Africa/Algiers		Canonical	+01:00	+01:00	
ER	+1520+03853	Africa/Asmara		Alias	+03:00	+03:00	Link to Africa/Nairobi
ML	+1239−00800	Africa/Bamako		Alias	+00:00	+00:00	Link to Africa/Abidjan
CF	+0422+01835	Africa/Bangui		Alias	+01:00	+01:00	Link to Africa/Lagos
GM	+1328−01639	Africa/Banjul		Alias	+00:00	+00:00	Link to Africa/Abidjan
GW	+1151−01535	Africa/Bissau		Canonical	+00:00	+00:00	
MW	−1547+03500	Africa/Blantyre		Alias	+02:00	+02:00	Link to Africa/Maputo
CG	−0416+01517	Africa/Brazzaville		Alias	+01:00	+01:00	Link to Africa/Lagos
BI	−0323+02922	Africa/Bujumbura		Alias	+02:00	+02:00	Link to Africa/Maputo
EG	+3003+03115	Africa/Cairo		Canonical	+02:00	+02:00	

If you do not know what Area and Location corresponds to your time zone, but you do know your offset from GMT, or your legacy time zone name (such as EST), scroll down in the table. There are TZ database names that correspond to these time zone identifiers, as shown in the following image.

		EST		Deprecated	−05:00	−05:00	Choose a zone that currently *observes* EST without daylight saving time, such as America/Cancun.
		EST5EDT		Deprecated	−05:00	−04:00	Choose a zone that *observes* EST with United States daylight saving time rules, such as America/New_York.
		Etc/GMT		Canonical	+00:00	+00:00	
		Etc/GMT+0		Alias	+00:00	+00:00	Link to Etc/GMT
		Etc/GMT+1		Canonical	−01:00	−01:00	Sign is intentionally inverted. See the Etc area description.
		Etc/GMT+10		Canonical	−10:00	−10:00	Sign is intentionally inverted. See the Etc area description.
		Etc/GMT+11		Canonical	−11:00	−11:00	Sign is intentionally inverted. See the Etc area description.
		Etc/GMT+12		Canonical	−12:00	−12:00	Sign is intentionally inverted. See the Etc area description.
		Etc/GMT+2		Canonical	−02:00	−02:00	Sign is intentionally inverted. See the Etc area description.
		Etc/GMT+3		Canonical	−03:00	−03:00	Sign is intentionally inverted. See the Etc area description.
		Etc/GMT+4		Canonical	−04:00	−04:00	Sign is intentionally inverted. See the Etc area description.
		Etc/GMT+5		Canonical	−05:00	−05:00	Sign is intentionally inverted. See the Etc area description.

Note: If you use a standard IANA time zone database name in the form "Area/Location" (for example, "America/New_York"), automatic adjustments are made for Daylight Savings Time. If you use a name that corresponds to an offset from GMT or to a legacy time zone name, it is your responsibility to account for Daylight Savings Time.

DT_TOUTC: Converting Local Time to UTC Time

DT_TOUTC takes a local date-time value and an IANA time zone name and converts the local time to UTC time.

Syntax: **How to Convert Local Time to UTC Time**

```
DT_TOUTC(datetime, timezone)
```

where:

datetime
Date-time

Is a date-time expression representing local time, containing date and time components.

timezone
Alphanumeric

Is a character expression containing the IANA time zone name of the local time, in the form 'Area/Location' (for example, 'America/New_York').

Example: **Converting Local Time to UTC Time**

The following request converts the current local date-time value for time zone America/New_York to UTC time.

```
TABLE FILE GGSALES
SUM DOLLARS NOPRINT
COMPUTE LOCAL1/HYYMDS  = DT_CURRENT_DATETIME(SECOND);
COMPUTE UTC1/HYYMDS = DT_TOUTC(LOCAL1, 'America/New_York');
ON TABLE SET PAGE NOLEAD
ON TABLE SET STYLE *
GRID=OFF,$
ENDSTYLE
END
```

The output is shown in the following image.

LOCAL1	UTC1
2020/09/04 14:49:41	2020/09/04 18:49:41

Example: **Sorting by UTC Time**

The following request retrieves the current date and time into the field LOCALT1 and sets the field TIMEZONE to IANA time zone database names. It then uses DT_TOUTC to convert the same local time, with different time zones, to the UTC time that corresponds to the given time zone, and sorts the output by the generated UTC time.

```
DEFINE FILE GGSALES
LOCALT1/HYYMDS=DT_CURRENT_DATETIME(SECOND);
TIMEZONE/A30=IF LAST TIMEZONE EQ ' ' THEN 'AMERICA/NEW_YORK'
 ELSE IF LAST TIMEZONE EQ 'AMERICA/NEW_YORK' THEN 'AMERICA/CHICAGO'
 ELSE IF LAST TIMEZONE EQ 'AMERICA/CHICAGO' THEN  'AMERICA/DENVER'
 ELSE IF LAST TIMEZONE EQ 'AMERICA/DENVER' THEN 'ASIA/TOKYO'
 ELSE IF LAST TIMEZONE EQ 'ASIA/TOKYO' THEN 'EUROPE/LONDON'
 ELSE IF LAST TIMEZONE EQ 'EUROPE/LONDON' THEN 'AMERICA/NEW_YORK';
UTCTIME/HYYMDS=DT_TOUTC(LOCALT1,TIMEZONE);
END
TABLE FILE GGSALES
PRINT TIMEZONE LOCALT1 DOLLARS NOPRINT
BY UTCTIME
WHERE PRODUCT EQ 'Thermos'
IF RECORDLIMIT EQ 20
ON TABLE SET PAGE NOLEAD
ON TABLE SET STYLE *
GRID=OFF,$
ENDSTYLE
END
```

The output is shown in the following image.

UTCTIME	TIMEZONE	LOCALT1
2020/10/02 06:45:59	ASIA/TOKYO	2020/10/02 15:45:59
	ASIA/TOKYO	2020/10/02 15:45:59
	ASIA/TOKYO	2020/10/02 15:45:59
	ASIA/TOKYO	2020/10/02 15:45:59
2020/10/02 14:45:59	EUROPE/LONDON	2020/10/02 15:45:59
	EUROPE/LONDON	2020/10/02 15:45:59
	EUROPE/LONDON	2020/10/02 15:45:59
	EUROPE/LONDON	2020/10/02 15:45:59
2020/10/02 19:45:59	AMERICA/NEW_YORK	2020/10/02 15:45:59
	AMERICA/NEW_YORK	2020/10/02 15:45:59
	AMERICA/NEW_YORK	2020/10/02 15:45:59
	AMERICA/NEW_YORK	2020/10/02 15:45:59
2020/10/02 20:45:59	AMERICA/CHICAGO	2020/10/02 15:45:59
	AMERICA/CHICAGO	2020/10/02 15:45:59
	AMERICA/CHICAGO	2020/10/02 15:45:59
	AMERICA/CHICAGO	2020/10/02 15:45:59
2020/10/02 21:45:59	AMERICA/DENVER	2020/10/02 15:45:59
	AMERICA/DENVER	2020/10/02 15:45:59
	AMERICA/DENVER	2020/10/02 15:45:59
	AMERICA/DENVER	2020/10/02 15:45:59

DT_TOLOCAL: Converting UTC Time to Local Time

DT_TOLOCAL takes a UTC date-time value and an IANA time zone name and converts the UTC time to local time.

Syntax: How to Convert UTC Time to Local Time

```
DT_TOLOCAL(datetime, timezone)
```

where:

datetime
: Date-time

  Is a date-time expression representing UTC time, containing date and time components.

timezone
Alphanumeric

Is a character expression containing the IANA time zone name of the local time, in the form 'Area/Location' (for example, 'America/New_York').

Example: **Converting UTC Time to Local Time**

The following request converts the current date-time value from UTC time to local time for time zone America/New_York.

```
TABLE FILE GGSALES
SUM DOLLARS NOPRINT
COMPUTE UTC1/HYYMDS  = DT_CURRENT_DATETIME(SECOND);
COMPUTE LOCAL1/HYYMDS = DT_TOLOCAL(UTC1, 'America/New_York');
ON TABLE SET PAGE NOLEAD
ON TABLE SET STYLE *
GRID=OFF,$
ENDSTYLE
END
```

The output is shown in the following image.

UTC1	LOCAL1
2020/09/04 15:00:26	2020/09/04 11:00:26

Support for Functions Used in ODBC Connector Client Tools

FOCUS and the SQL translator now support the following functions used by client tools with the ODBC Connector.

FOCUS and SQL Functions

Function Name	Description	Syntax
ASCII	Returns the ASCII code value of the leftmost character of a character expression.	ASCII(*charexp*) For example, the following returns the value 65. ASCII('A')

Function Name	Description	Syntax

DAYNAME	Returns a character string that contains the data-source-specific name of the day for the day part of a date expression.	`DAYNAME(date_exp)`
		For example, the following returns Monday:
		`DAYNAME('August 3, 2020')`

DIFFERENCE	Returns an integer value measuring the difference between the SOUNDEX or METAPHONE values of two different character expressions. Zero (0) represents the least similarity. For SOUNDEX, 4 represents the most similarity, and for METAPHONE, 16 represents the most similarity.
	The use of SOUNDEX or METAPHONE depends on the PHONETIC_ALGORITHM setting. METAPHONE is the default algorithm.
		`DIFFERENCE(chrexp1, chrexp2)`
		For example, the following returns the value 4 when SOUNDEX is the phonetic algorithm:
		`DIFFERENCE('Green','Greene')`

LEFT	Given a character string, or an expression that can be converted to varchar, and an integer number, returns that number of characters from the left of the string.	`LEFT(chr_exp, int_exp)`
		For example, the following returns the value *ab*:
		`LEFT('abcdefg',2)`

LOG10	Returns the base-10 logarithm of a numeric expression.	`LOG10(num_exp)`
		For example, the following returns the value 2.161:
		`LOG10(145)`

Function Name	Description	Syntax

MONTHNAME	Returns a character string that contains the data-source-specific name of the month for the month part of a date expression.	`MONTHNAME(date_exp)`
		For example, the following returns August:
		`MONTHNAME('August 3, 2020')`

OVERLAY	Given a starting position, length, source string, and insertion string, replaces the number of characters defined by *length* in the source string with the insertion string, starting from the starting position.	`OVERLAY(src, ins, start, len)`
		For example, the following returns SCOTLAND by replacing the first 3 characters in ENGLAND with the characters *SCOT*:
		`OVERLAY('ENGLAND', 'SCOT', 1, 3)`

POSITION	Given a search string, a source string, and a starting position, returns the position of the search string within the source string. The search starts at the given starting position. If the string is not found, returns zero (0). The search is case sensitive.	`POSITION(search, source, start)`
		For example, when CustomerName is Sandra Arzola, the following returns 8:
		`POSITION('A', CustomerName, 3)`

REPEAT	Given a source string and an integer number, returns a string with the source string repeated that number of times, each repetition separated from the previous one with a space.	`REPEAT(source_str, number)`
		For example, when FIRST_NAME is MARY, the following returns the string *MARY MARY MARY*:
		`REPEAT(FIRST_NAME, 3)`

RIGHT	Given a character string, or an expression that can be converted to varchar, and an integer number, returns that number of characters from the right of the string.	`RIGHT(char_exp, integer_exp)`
		For example, the following returns the value *fg*:
		`RIGHT('abcdefg',2)`

Function Name	Description	Syntax

ROUND	Given a numeric expression and an integer count, returns the numeric expression rounded to that number of decimal places. If the number of decimal places is negative, it rounds to the left of the decimal point.	`ROUND(num_exp, count)`
		For example, the following returns 1.23500.
		`ROUND(1.23456, 3)`

SIGN	Given a numeric expression, returns the value 1 if it is positive, or -1 if it is negative. SIGN(0) returns 0.	`SIGN(num_exp)`
		For example, the following returns 1.
		`SIGN(1.23456)`

SPACE	Given an integer count, returns a string consisting of that number of spaces.	`SPACE(count)`
		For example, the following returns a string consisting of two spaces.
		`SPACE(2)`

TRUNCATE	Truncates a numeric expression to a given number of decimal places. If the number of decimal places is negative, the number is truncated to the left of the decimal point.	`TRUNCATE(num_exp, count)`
		For example, the following returns 1.23400.
		`TRUNCATE(1.23456, 3)`

SQL Functions

Function Name	Description	Syntax

CHR	Takes a number as an argument and returns the ASCII character.	`CHR(number)`
		For example, the following returns ¾.
		`CHR(190)`

Function Name	Description	Syntax

LOCATE	Given a substring, a source string and a starting position (the default is 1), returns the position of the first occurrence of the substring, starting the search at the starting position. If the substring is not found, returns zero (0). The search is case insensitive.	`LOCATE(substr, source [,start])`
		For example, when CustomerName is Sandra Arzola, the following returns 6:
		`LOCATE('a', CustomerName, 3)`
		The following returns 2:
		`LOCATE('a', CustomerName)`

IMPUTE: Replacing Missing Values With Aggregated Values

IMPUTE calculates a value to replace missing numeric data on report output, within a partition.

In place of eliminating data records with missing values from analysis, IMPUTE enables you to substitute a variety of estimates for the missing values, including the mean, the median, the mode, or a numeric constant, all calculated within the data partition specified by the reset key. This function is designed to be used with detail level reports (PRINT or LIST commands), and with calculated values (fields created with the COMPUTE command).

The syntax is:

`IMPUTE(field, reset_key, replacement)`

where:

field
: Is the name of the numeric input field that is defined with MISSING ON.

reset_key
: Defines the partition for the calculation. Valid values are:

- A sort field name.
- PRESET, which uses the break defined by the SET PARTITION_ON command.
- TABLE, which performs the calculation on the entire table.

replacement

Is a numeric constant or one of the following:

- MEAN
- MEDIAN
- MODE

Example: **Replacing Missing Values With Aggregated Values**

To run this example, the FOCUS data source SALEMISS must be created. SALEMISS is the SALES data source with some missing values added in the RETURNS and DAMAGED fields. The following is the SALEMISS Master File, which should be added to the IBISAMP application.

```
FILENAME=KSALES, SUFFIX=FOC, REMARKS='Legacy Metadata Sample: sales',$

SEGNAME=STOR_SEG, SEGTYPE=S1,
   FIELDNAME=STORE_CODE,  ALIAS=SNO,  FORMAT=A3,    $
   FIELDNAME=CITY,        ALIAS=CTY,  FORMAT=A15,   $
   FIELDNAME=AREA,        ALIAS=LOC,  FORMAT=A1,    $

SEGNAME=DATE_SEG, PARENT=STOR_SEG, SEGTYPE=SH1,
   FIELDNAME=DATE,        ALIAS=DTE,  FORMAT=A4MD, $

SEGNAME=PRODUCT, PARENT=DATE_SEG, SEGTYPE=S1,
   FIELDNAME=PROD_CODE,     ALIAS=PCODE,   FORMAT=A3,     FIELDTYPE=I, $
   FIELDNAME=UNIT_SOLD,     ALIAS=SOLD,    FORMAT=I5,     $
   FIELDNAME=RETAIL_PRICE,  ALIAS=RP,      FORMAT=D5.2M,  $
   FIELDNAME=DELIVER_AMT,   ALIAS=SHIP,    FORMAT=I5,     $
   FIELDNAME=OPENING_AMT,   ALIAS=INV,     FORMAT=I5,     $
   FIELDNAME=RETURNS,       ALIAS=RTN,     FORMAT=I3,     MISSING=ON, $
   FIELDNAME=DAMAGED,       ALIAS=BAD,     FORMAT=I3,     MISSING=ON, $
```

The following procedure creates the SALEMISS data source and then adds the missing values to the RETURNS and DAMAGED fields:

```
CREATE FILE ibisamp/SALEMISS
MODIFY FILE ibisamp/SALEMISS
FIXFORM STORE_CODE/3 CITY/15 AREA/1 DATE/4 PROD_CODE/3
FIXFORM UNIT_SOLD/5 RETAIL_PRICE/5 DELIVER_AMT/5
FIXFORM OPENING_AMT/5 RETURNS/3 DAMAGED/3
MATCH STORE_CODE
ON NOMATCH INCLUDE
ON MATCH CONTINUE
MATCH DATE
ON NOMATCH INCLUDE
ON MATCH CONTINUE
MATCH PROD_CODE
ON NOMATCH INCLUDE
ON MATCH REJECT
DATA
14BSTAMFORD       S1212B10   60  .95   80   65 10  6
14BSTAMFORD       S1212B12   40 1.29   20   50  3  3
14BSTAMFORD       S1212B17   29 1.89   30   30  2  1
14BSTAMFORD       S1212C13   25 1.99   30   40  3  0
14BSTAMFORD       S1212C7    45 2.39   50   49  5  4
14BSTAMFORD       S1212D12   27 2.19   40   35  0  0
14BSTAMFORD       S1212E2    80  .99  100  100  9  4
14BSTAMFORD       S1212E3    70 1.09   80   90  8  9
14ZNEW YORK       U1017B10   30  .85   30   10  2  3
14ZNEW YORK       U1017B17   20 1.89   40   25  2  1
14ZNEW YORK       U1017B20   15 1.99   30    5  0  1
14ZNEW YORK       U1017C17   12 2.09   10   15  0  0
14ZNEW YORK       U1017D12   20 2.09   30   10  3  2
14ZNEW YORK       U1017E1    30  .89   25   45  4  7
14ZNEW YORK       U1017E3    35 1.09   25   45  4  2
77FUNIONDALE      R1018B20   25 2.09   40   25  1  1
77FUNIONDALE      R1018C7    40 2.49   40   40  0  0
K1 NEWARK         U1019B12   29 1.49   30   30  1  0
K1 NEWARK         U1018B10   13  .99   30   15  1  1
END
-RUN
```

```
MODIFY FILE ibisamp/SALEMISS
FIXFORM STORE_CODE/3 DATE/5 PROD_CODE/4
FIXFORM UNIT/3 RETAIL/5 DELIVER/3
FIXFORM OPEN/3 RETURNS/C3 DAMAGED/C3
MATCH STORE_CODE
ON NOMATCH INCLUDE
ON MATCH CONTINUE
MATCH DATE
ON NOMATCH INCLUDE
ON MATCH CONTINUE
MATCH PROD_CODE
ON NOMATCH INCLUDE
ON MATCH REJECT
DATA
14Z1017 C13 15 1.99 35 30    6
14Z1017 C14 18 2.05 30 25 4
14Z1017 E2  33 0.99 45 40
END
-RUN
```

The following request against the SALEMISS data source generates replacement values for the missing values in the RETURNS field, using only the values within the same store.

```
SET PARTITION_ON=FIRST
TABLE FILE SALEMISS
PRINT RETURNS
COMPUTE MEDIAN1 = IMPUTE(RETURNS, PRESET, MEDIAN);
COMPUTE MEAN1 = IMPUTE(RETURNS, PRESET, MEAN);
COMPUTE MODE1 = IMPUTE(RETURNS, PRESET, MODE);
BY STORE_CODE
ON TABLE SET PAGE NOPAGE
ON TABLE SET STYLE *
TYPE=REPORT, GRID=OFF,$
ENDSTYLE
END
```

The output is shown in the following image. The missing values occur in store 14Z, and the replacement values are calculated using only the RETURNS values from that store because PARTITION_ON is set to FIRST.

STORE_CODE	RETURNS	MEDIAN1	MEAN1	MODE1
14B	10	10.00	10.00	10.00
	3	3.00	3.00	3.00
	2	2.00	2.00	2.00
	3	3.00	3.00	3.00
	5	5.00	5.00	5.00
	0	.00	.00	.00
	9	9.00	9.00	9.00
	8	8.00	8.00	8.00
14Z	2	2.00	2.00	2.00
	2	2.00	2.00	2.00
	0	.00	.00	.00
	.	2.00	2.00	4.00
	4	4.00	4.00	4.00
	0	.00	.00	.00
	3	3.00	3.00	3.00
	4	4.00	4.00	4.00
	.	2.00	2.00	4.00
	4	4.00	4.00	4.00
77F	1	1.00	1.00	1.00
	0	.00	.00	.00
K1	1	1.00	1.00	1.00
	1	1.00	1.00	1.00

Changing the PARTITION_ON setting to TABLE produces the following output, in which the replacement values are calculated using all of the rows in the table.

STORE_CODE	RETURNS	MEDIAN1	MEAN1	MODE1
14B	10	10.00	10.00	10.00
	3	3.00	3.00	3.00
	2	2.00	2.00	2.00
	3	3.00	3.00	3.00
	5	5.00	5.00	5.00
	0	.00	.00	.00
	9	9.00	9.00	9.00
	8	8.00	8.00	8.00
14Z	2	2.00	2.00	2.00
	2	2.00	2.00	2.00
	0	.00	.00	.00
	.	2.00	3.00	.00
	4	4.00	4.00	4.00
	0	.00	.00	.00
	3	3.00	3.00	3.00
	4	4.00	4.00	4.00
	.	2.00	3.00	.00
	4	4.00	4.00	4.00
77F	1	1.00	1.00	1.00
	0	.00	.00	.00
K1	1	1.00	1.00	1.00
	1	1.00	1.00	1.00

OUTLIER: Identifying Outliers in Numeric Data

The 1.5 * IQR rule (where IQR means Inner Quartile Range) is a common way to identify outliers in data. This rule defines an outlier as a value that is above or below 1.5 times the inner quartile range in the data. The inner quartile range is based on sorting the data values, dividing it into equal quarters, and calculating the range of values between the first quartile (the value one quarter of the way through the sorted data) and third quartile (the value three quarters of the way through the sorted data). The value that is 1.5 times below the inner quartile range is called the *lower fence*, and the value that is 1.5 times above the inner quartile range is called the *upper fence*.

Given a numeric field as input, OUTLIER returns one of the following values for each value of the field, using the 1.5 * IQR rule:

- **0 (zero).** The value is not an outlier.
- **-1.** The value is below the lower fence.
- **1.** The value is above the upper fence.

Syntax: How to Identify Outliers in Numeric Data

```
OUTLIER(input_field)
```

where:

input_field
Numeric

Is the numeric field to be analyzed.

Example: Identifying Outliers

The following request defines the SALES field to have different values depending on the store code, and uses OUTLIER to determine whether each field value is an outlier.

```
DEFINE FILE GGSALES
SALES/D12 = IF ((CATEGORY EQ 'Coffee') AND (STCD EQ 'R1019')) THEN 19000
  ELSE IF ((CATEGORY EQ 'Coffee') AND (STCD EQ 'R1020')) THEN 20000
  ELSE IF ((CATEGORY EQ 'Coffee') AND (STCD EQ 'R1040')) THEN 7000
  ELSE DOLLARS;
END
TABLE FILE GGSALES
SUM SALES
COMPUTE OUT1/I3 = OUTLIER(SALES);
BY CATEGORY
BY STCD
WHERE CATEGORY EQ 'Coffee'
ON TABLE SET PAGE NOLEAD
ON TABLE SET STYLE *
GRID=OFF,$
ENDSTYLE
END
```

The output is shown in the following image. Values above 2 million are above the upper fence, values below 1 million are below the lower fence, and other values are not outliers:

Category	Store ID	SALES	OUT1
Coffee	R1019	2,280,000	1
	R1020	2,400,000	1
	R1040	840,000	-1
	R1041	1,576,915	0
	R1044	1,340,437	0
	R1088	1,375,040	0
	R1100	1,364,420	0
	R1109	1,459,160	0
	R1200	1,463,453	0
	R1244	1,553,962	0
	R1248	1,535,631	0
	R1250	1,386,124	0

Enhancement to the PARTITION_AGGR Function

The post-aggregation calculations MEDIAN and MODE have been added to the PARTITION_AGGR function.

Support for Standard Deviation in PARTITION_AGGR

The PARTITION_AGGR function generates rolling calculations based on a block of rows from the internal matrix of a TABLE request. Population Standard Deviation (STDP) and Sample Standard Deviation (STDS) have been added as operations for the rolling calculation.

Note: Using the STDS or STDP aggregation operators requires that the request use the PRINT display command to avoid duplicate aggregation steps.

The syntax is:

```
PARTITION_AGGR([prefix.]measure,reset_key,lower,upper,operation)
```

where:

prefix.
Defines an aggregation operator to apply to the measure before using it in the rolling calculation. Valid operators are:

- **SUM.** which calculates the sum of the measure field values. SUM is the default operator.
- **CNT.** which calculates a count of the measure field values.
- **AVE.** which calculates the average of the measure field values.
- **MIN.** which calculates the minimum of the measure field values.
- **MAX.** which calculates the maximum of the measure field values.
- **FST.** which retrieves the first value of the measure field.
- **LST.** which retrieves the last value of the measure field.
- **STDP.** which retrieves the population standard deviation of the measure field.
- **STDS.** which retrieves the sample standard deviation of the measure field.

Note: The operators PCT., RPCT., TOT., MDN., and DST. are not supported. COMPUTEs that reference those unsupported operators are also not supported.

measure

Is the measure field to be aggregated. It can be a real field in the request or a calculated value generated with the COMPUTE command, as long as the COMPUTE does not reference an unsupported prefix operator.

reset_key

Identifies the point at which the calculation restarts. Valid values are:

- The name of a sort field in the request.
- PRESET, which uses the value of the PARTITION_ON parameter, as described in the *Using Functions* manual.
- TABLE, which indicates that there is no break on a sort field.

The sort field may use BY HIGHEST to indicate a HIGH-TO-LOW sort. ACROSS COLUMNS AND is supported. BY ROWS OVER and FOR are not supported.

lower

Identifies the starting point for the rolling calculation. Valid values are:

- **n, -n**, which starts the calculation *n* rows forward or back from the current row.
- **B**, which starts the calculation at the beginning of the current sort break (the first line with the same sort field value as the current line).

upper

Identifies the ending point of the rolling calculation. The *lower* row value must precede *upper* row value.

Valid values are:

- **C**, which ends the rolling calculation at the current row in the internal matrix.
- **n, -n**, which ends the calculation *n* rows forward or back from the current row.
- **E**, which ends the rolling calculation at the end of the sort break (the last line with the same sort value as the current row.)

Note: The values used in the calculations depend on the sort sequence (ascending or descending) specified in the request. Be aware that displaying a date or time dimension in descending order may produce different results than those you may expect.

operation

Specifies the rolling calculation used on the values in the internal matrix. The new supported operations are:

- **STDP.** which calculates a population standard deviation.
- **STDS.** which calculates a sample standard deviation.

Example: **Using PARTITION_AGGR to Calculate a Population Standard Deviation**

The following request uses the STDP aggregation operator in PARTITION_AGGR to calculate the standard deviation for each category.

```
TABLE FILE ggsales
PRINT DOLLARS
COMPUTE STDP1/D12.2M = PARTITION_AGGR(DOLLARS, CATEGORY, B, E, STDP);
BY CATEGORY
BY PRODUCT
ON TABLE SET PAGE NOLEAD
ON TABLE SET STYLE *
GRID=OFF,$
ENDSTYLE
END
```

Partial output is shown in the following image.

Category	Product	Dollar Sales	STDP1
Coffee	Capuccino	20805	$6,358.13
		20748	$6,358.13
		20376	$6,358.13
		20028	$6,358.13
		19905	$6,358.13
Food	Biscotti	18200	$6,565.19
		18084	$6,565.19
		17100	$6,565.19
		16918	$6,565.19
		16656	$6,565.19
Gifts	Coffee Grinder	7752	$4,518.06
		7715	$4,518.06
		7623	$4,518.06
		7485	$4,518.06

...

...

Output Format Enhancements

This topic describes enhancements for output formats generated by the HOLD, SAVE, and SAVB commands.

Generating Format XLSX Worksheets in FOCUS

In prior releases of FOCUS, you needed to use the SET EXCELSERVURL parameter to a WebFOCUS servlet in order to generate FORMAT XLSX output. Now, you can use the following SET EXCELSERVURL command to generate the output within FOCUS, without needing an installation of WebFOCUS.

```
SET EXCELSERVURL = LOCAL
```

You can place this command in a supported profile, in a FOCEXEC, or in a TABLE request (using the syntax ON TABLE SET EXCELSERVURL LOCAL).

Defining a Hyperlink Color for a Report Component

You can use the HYPERLINK-COLOR attribute in a StyleSheet to designate a color for a hyperlink within a report. This applies to all hyperlinks generated in the report. You can define a single color for the entire report or different colors for each individual component.

Syntax: How to Define a Hyperlink Color

```
TYPE = type, HYPERLINK-COLOR = color
```

where:

type
: Is the report component you wish to affect. You can apply this keyword to the entire report using TYPE=REPORT. The attribute can also individually be set for any other element of the report.

color
: Can use any style sheet supported color value designation.

Example: Defining a Hyperlink Color

The following PDF request illustrates how to define hyperlink colors for the entire report, as well as individual elements.

- The default font color for the entire report is grey and the default hyperlink color for the entire report is slate blue.
- For the Dollar Sales column (DOLLARS), the font color is green and the hyperlink color is purple.
- For both the Dollar Sales column (DOLLARS) and the Unit Sales column (UNITS),conditional styling has been applied using the same condition (REGION GE 'O').
- For the Unit Sales column (UNITS), when the conditional styling is met, the hyperlink color is inherited from the default hyperlink color for the report (slate blue).
- For the Dollar Sales column (DOLLARS), when the conditional styling is met, the hyperlink color is purple.

```
TABLE FILE GGSALES
SUM DOLLARS/D12CM UNITS/D12C
BY REGION
BY CATEGORY
HEADING
"Hyperlinks of Many Colors"
""
ON TABLE SET PAGE-NUM OFF
ON TABLE HOLD AS PDFHYP FORMAT PDF
ON TABLE SET STYLE *
TYPE=REPORT, SQUEEZE=ON, FONT=ARIAL, GRID=OFF, COLOR=GREY,
HYPERLINK-COLOR='SLATE BLUE',$
TYPE=DATA, COLUMN=UNITS, WHEN=REGION GE 'O', URL='http://www.tibco.com',$
TYPE=DATA, COLUMN=DOLLARS, COLOR=GREEN, HYPERLINK-COLOR='PURPLE',$
TYPE=DATA, COLUMN=DOLLARS, WHEN=REGION GE 'O', URL='http://www.tibco.com',$
ENDSTYLE
END
```

The output is:

Hyperlinks of Many Colors

Region	Category	Dollar Sales	Unit Sales
Midwest	Coffee	$4,178,513	332,777
	Food	$4,404,483	346,421
	Gifts	$2,931,349	234,463
Northeast	Coffee	$4,201,057	339,155
	Food	$4,445,197	357,919
	Gifts	$2,848,289	227,529
Southeast	Coffee	$4,435,134	352,357
	Food	$4,308,731	349,829
	Gifts	$3,037,420	237,928
West	Coffee	$4,493,483	358,426
	Food	$4,204,333	340,367
	Gifts	$2,977,092	235,042

Inserting Text and Images Into XLSX Workbook Headers and Footers

FOCUS supports the insertion of text and images into Excel headers and footers and the definition of key page settings to support the placement of text and images in relationship to the overall worksheet and the Excel generated page breaks. This access to the Excel page functionality is designed to enhance overall usability of the worksheets for users who will be printing these reports. Page settings including orientation, page size, and page margins will directly affect the layout of each Excel page based on values defined within the FOCEXEC. Images and text can be included on headers and footers on every printed page, on the first page of the report only, or only on all subsequent pages. The FOCUS headings and footings continue to display within the worksheet. With this feature, FOCUS can insert logos to be printed once at the top of a report and watermark images that need to be displayed on every printed page.

Syntax: How to Insert Text and Images Into XLSX Workbook Headers and Footers

To place images in XLSX Workbook headers and footers, the syntax is:

```
TYPE={PAGEHEADER|PAGEFOOTER},OBJECT=IMAGE,
 IMAGE=imagename, JUSTIFY={LEFT|CENTER|RIGHT}
 [,DISPLAYON={FIRST|NOT-FIRST}] [,SIZE=(w h)],$
```

To place text in XLSX Workbook headers and footers, the syntax is:

```
TYPE={PAGEHEADER|PAGEFOOTER},OBJECT=STRING,
 TEXT=text, JUSTIFY={LEFT|CENTER|RIGHT}
 [,DISPLAYON={FIRST|NOT-FIRST}] ,$
```

where:

`PAGEHEADER`
: Places the text or image in the worksheet header.

`PAGEFOOTER`
: Places the text or image in the worksheet footer.

`imagename`
: Is the name of a valid image file to be placed in the header or footer. The image must be located in the defined application path or allocated to a DDNAME accessible to FOCUS. The image types supported are GIF and JPEG.

`text`
: Is the text to be placed in the header or footer.

`JUSTIFY={LEFT|CENTER|RIGHT}`
: Identifies the area in the header or footer to contain the text or image and the justification or placement within that defined area.

DISPLAYON
Defines whether the text or image should be placed on the first page only or on all pages except the first. Omit this attribute to place the text or image on all pages.

Valid values are:

- FIRST places the text or image only on the first page.
- NOT-FIRST places the text or image on every page, except the first page.

SIZE=(*w h*)
Is the size of the image. By default, an image is added at its original size.

w is the width of the image, expressed in the unit of measurement specified by the UNITS parameter.

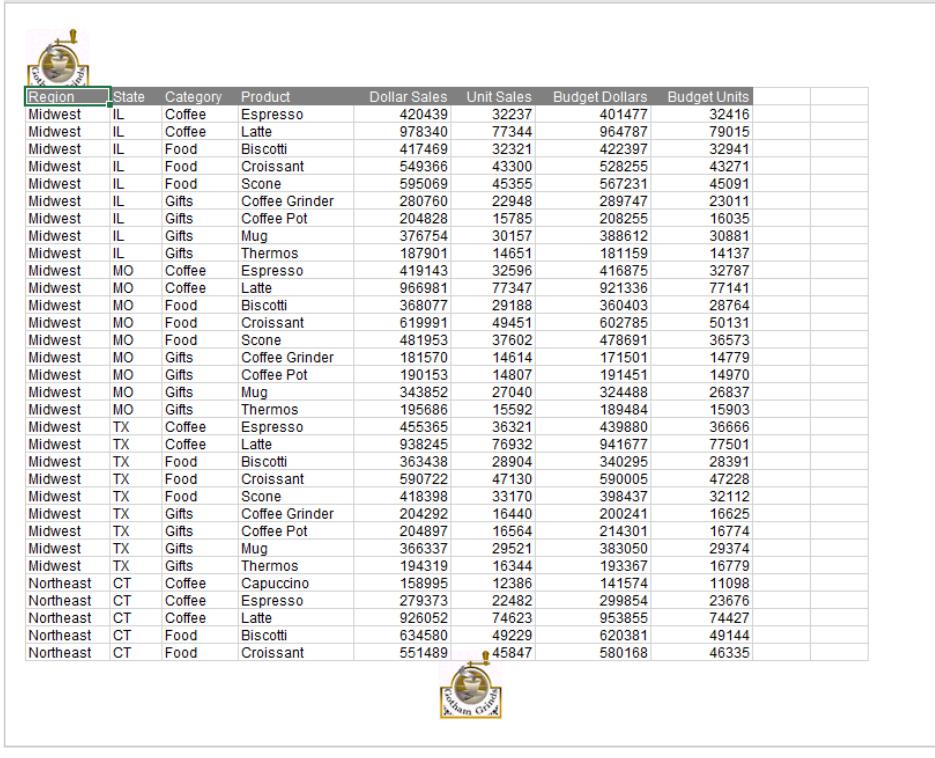
h is the height of the image, expressed in the unit of measurement specified by the UNITS parameter.

Example: Inserting Images in Excel Headers and Footers and Defining Page Settings

The following request against the GGSALES data source places the image gglogo.gif on the left header area of the first page and the right header area of every subsequent page of the resulting worksheet. It places the same image in the center area of the footer on every page. The image file is in a data set allocated to DDNAME GIF.

```
SET EXCELSERVURL = LOCAL
TABLE FILE GGSALES
SUM DOLLARS UNITS BUDDOLLARS BUDUNITS
BY REGION
BY ST
BY CATEGORY
BY PRODUCT
ON TABLE SET BYDISPLAY ON
ON TABLE HOLD AS XLSXHD1 FORMAT XLSX
ON TABLE SET STYLE *
FONT=ARIAL,SIZE=12,
XLSXPAGESETS=ON,TOPMARGIN=1,BOTTOMMARGIN=1,ORIENTATION=LANDSCAPE,
PAGESIZE=LETTER,$
TYPE=TITLE, COLOR=WHITE, BACKCOLOR=GREY,$
TYPE=PAGEHEADER, OBJECT=IMAGE, JUSTIFY=LEFT, IMAGE=GGLOGO.GIF,
DISPLAYON=FIRST,$
TYPE=PAGEHEADER, OBJECT=IMAGE, JUSTIFY=RIGHT, IMAGE=GGLOGO.GIF,
DISPLAYON=NOT-FIRST,$
TYPE=PAGEFOOTER, OBJECT=IMAGE, JUSTIFY=CENTER, IMAGE=GGLOGO.GIF,$
END
```

The first page of output has the image in the left area of the header and in the center area of the footer, as shown in the following image.



Region	State	Category	Product	Dollar Sales	Unit Sales	Budget Dollars	Budget Units
Midwest	IL	Coffee	Espresso	420439	32237	401477	32416
Midwest	IL	Coffee	Latte	978340	77344	964787	79015
Midwest	IL	Food	Biscotti	417469	32321	422397	32941
Midwest	IL	Food	Croissant	549366	43300	528255	43271
Midwest	IL	Food	Scone	595069	45355	567231	45091
Midwest	IL	Gifts	Coffee Grinder	280760	22948	289747	23011
Midwest	IL	Gifts	Coffee Pot	204828	15785	208255	16035
Midwest	IL	Gifts	Mug	376754	30157	388612	30881
Midwest	IL	Gifts	Thermos	187901	14651	181159	14137
Midwest	MO	Coffee	Espresso	419143	32596	416875	32787
Midwest	MO	Coffee	Latte	966981	77347	921336	77141
Midwest	MO	Food	Biscotti	368077	29188	360403	28764
Midwest	MO	Food	Croissant	619991	49451	602785	50131
Midwest	MO	Food	Scone	481953	37602	478691	36573
Midwest	MO	Gifts	Coffee Grinder	181570	14614	171501	14779
Midwest	MO	Gifts	Coffee Pot	190153	14807	191451	14970
Midwest	MO	Gifts	Mug	343852	27040	324488	26837
Midwest	MO	Gifts	Thermos	195686	15592	189484	15903
Midwest	TX	Coffee	Espresso	455365	36321	439880	36666
Midwest	TX	Coffee	Latte	938245	76932	941677	77501
Midwest	TX	Food	Biscotti	363438	28904	340295	28391
Midwest	TX	Food	Croissant	590722	47130	590005	47228
Midwest	TX	Food	Scone	418398	33170	398437	32112
Midwest	TX	Gifts	Coffee Grinder	204292	16440	200241	16625
Midwest	TX	Gifts	Coffee Pot	204897	16564	214301	16774
Midwest	TX	Gifts	Mug	366337	29521	383050	29374
Midwest	TX	Gifts	Thermos	194319	16344	193367	16779
Northeast	CT	Coffee	Capuccino	158995	12386	141574	11098
Northeast	CT	Coffee	Espresso	279373	22482	299854	23676
Northeast	CT	Coffee	Latte	926052	74623	953855	74427
Northeast	CT	Food	Biscotti	634580	49229	620381	49144
Northeast	CT	Food	Croissant	551489	45847	580168	46335

The second page of output has the image in the right area of the header and in the center area of the footer, as shown in the following image.

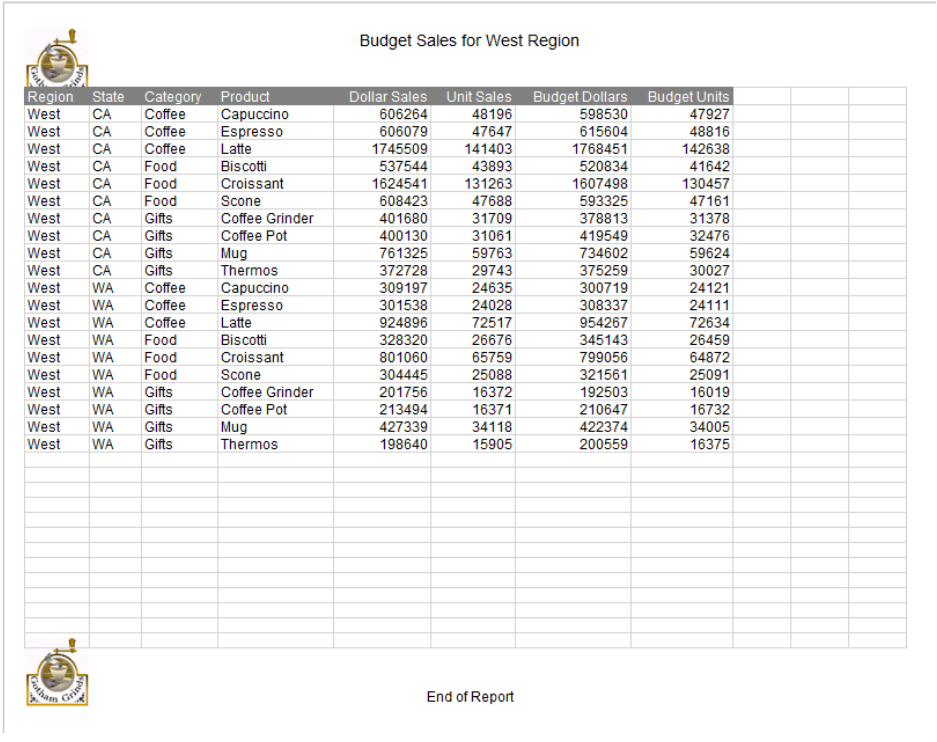
Northeast	CT	Food	Scone	283874	22378	269221	21038		
Northeast	CT	Gifts	Coffee Grinder	169908	13691	159620	13117		
Northeast	CT	Gifts	Coffee Pot	208209	15523	197051	15190		
Northeast	CT	Gifts	Mug	392967	31728	424333	32415		
Northeast	CT	Gifts	Thermos	221827	17568	219025	17667		
Northeast	MA	Coffee	Capuccino	174344	15358	192747	15672		
Northeast	MA	Coffee	Espresso	248356	19698	254310	19888		
Northeast	MA	Coffee	Latte	917737	74572	941438	73874		
Northeast	MA	Food	Biscotti	570391	47064	616766	48246		
Northeast	MA	Food	Croissant	497234	41029	519322	41351		
Northeast	MA	Food	Scone	332486	25363	312004	23774		
Northeast	MA	Gifts	Coffee Grinder	177940	14382	187686	15384		
Northeast	MA	Gifts	Coffee Pot	184119	15349	184071	15171		
Northeast	MA	Gifts	Mug	401944	32360	401617	31324		
Northeast	MA	Gifts	Thermos	203435	16734	208436	16921		
Northeast	NY	Coffee	Capuccino	208756	17041	227170	17662		
Northeast	NY	Coffee	Espresso	322378	25947	318738	26212		
Northeast	NY	Coffee	Latte	965066	77048	964733	76528		
Northeast	NY	Food	Biscotti	662237	53500	658781	51808		
Northeast	NY	Food	Croissant	622095	50518	640032	50178		
Northeast	NY	Food	Scone	290811	22991	284478	23603		
Northeast	NY	Gifts	Coffee Grinder	161352	12904	164336	12796		
Northeast	NY	Gifts	Coffee Pot	198452	15313	192227	15043		
Northeast	NY	Gifts	Mug	349300	27409	344364	26801		
Northeast	NY	Gifts	Thermos	178836	14568	187786	15179		
Southeast	FL	Coffee	Capuccino	317027	24143	285194	23092		
Southeast	FL	Coffee	Espresso	256539	19730	236531	18690		
Southeast	FL	Coffee	Latte	889887	71123	886465	72975		
Southeast	FL	Food	Biscotti	511597	40606	516984	41242		
Southeast	FL	Food	Croissant	602076	50175	644884	51437		
Southeast	FL	Food	Scone	311836	24543	299547	24576		
Southeast	FL	Gifts	Coffee Grinder	268384	20441	247445	20340		
Southeast	FL	Gifts	Coffee Pot	212057	16145	215467	16470		

Example: **Inserting Text and Images in Excel Report Output**

The following request against the GGSALES data source places the gglogo.gif image in the left header area and text in the center header area of the worksheet. It also places the image in the left footer area and text in the center footer area. The image is in a data set allocated to DDNAME GIF.

```
SET EXCELSERVURL=LOCAL
TABLE FILE GGSALES
SUM DOLLARS UNITS BUDDOLLARS BUDUNITS
BY REGION
BY ST
BY CATEGORY
BY PRODUCT
WHERE REGION EQ 'West'
ON TABLE SET BYDISPLAY ON
ON TABLE HOLD AS XLSXHD2 FORMAT XLSX
ON TABLE SET STYLE *
FONT=ARIAL,SIZE=12,
XLSXPAGESETS=ON,TOPMARGIN=1,BOTTOMMARGIN=1,ORIENTATION=LANDSCAPE,
PAGESIZE=LETTER,$
TYPE=TITLE, COLOR=WHITE, BACKCOLOR=GREY,$
TYPE=PAGEHEADER, OBJECT=IMAGE, JUSTIFY=LEFT, IMAGE=GGLOGO.GIF,$
TYPE=PAGEHEADER, OBJECT=STRING, JUSTIFY=CENTER,
  TEXT=Budget Sales for West Region, FONT=ARIAL, SIZE=14,$
 TYPE=PAGEFOOTER, OBJECT=IMAGE, JUSTIFY=LEFT, IMAGE=GGLOGO.GIF,$
 TYPE=PAGEFOOTER, OBJECT=STRING, JUSTIFY=CENTER,
  TEXT=End of Report, FONT=ARIAL, SIZE=12,$
 ENDSTYLE
 END
```

The output is shown in the following image.



Budget Sales for West Region

Region	State	Category	Product	Dollar Sales	Unit Sales	Budget Dollars	Budget Units
West	CA	Coffee	Capuccino	606264	48196	598530	47927
West	CA	Coffee	Espresso	606079	47647	615604	48816
West	CA	Coffee	Latte	1745509	141403	1768451	142638
West	CA	Food	Biscotti	537544	43893	520834	41642
West	CA	Food	Croissant	1624541	131263	1607498	130457
West	CA	Food	Scone	608423	47688	593325	47161
West	CA	Gifts	Coffee Grinder	401680	31709	378813	31378
West	CA	Gifts	Coffee Pot	400130	31061	419549	32476
West	CA	Gifts	Mug	761325	59763	734602	59624
West	CA	Gifts	Thermos	372728	29743	375259	30027
West	WA	Coffee	Capuccino	309197	24635	300719	24121
West	WA	Coffee	Espresso	301538	24028	308337	24111
West	WA	Coffee	Latte	924896	72517	954267	72634
West	WA	Food	Biscotti	328320	26676	345143	26459
West	WA	Food	Croissant	801060	65759	799056	64872
West	WA	Food	Scone	304445	25088	321561	25091
West	WA	Gifts	Coffee Grinder	201756	16372	192503	16019
West	WA	Gifts	Coffee Pot	213494	16371	210647	16732
West	WA	Gifts	Mug	427339	34118	422374	34005
West	WA	Gifts	Thermos	198640	15905	200559	16375

End of Report

Reference: **Usage Notes for Inserting Images Into XLSX Worksheet Headers and Footers**

- The Excel headers and footers are not automatically sized based on contents of the areas.
- Define page margins within the page settings (XLSPAGESETS) to account for the space required to display the images within each page of the report.
- The image sizing based on the specified height and width is not proportional. Sizing may cause image distortion.

Synchronizing FOCUS Page Breaks With Excel Page Breaks

FOCUS page breaks in format XLSX report output are now synchronized with Excel page breaks.

Controlling The Synchronization of FOCUS Page Breaks With Excel Page Breaks

The SET parameter XLSXPAGEBRKIGNORE controls whether page breaks in FOCUS format XLSX report output insert Excel page breaks at the same location in the output.

Syntax: How to Control Synchronization of FOCUS Page Breaks With Excel Page Breaks

In a procedure or profile, use the following syntax.

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

In a request, use the following syntax.

```
SET XLSXPAGEBRKIGNORE {OFF|ON}
```

where:

OFF
: Synchronizes FOCUS page breaks with Excel page breaks in format XLSX report output. This is the default value.

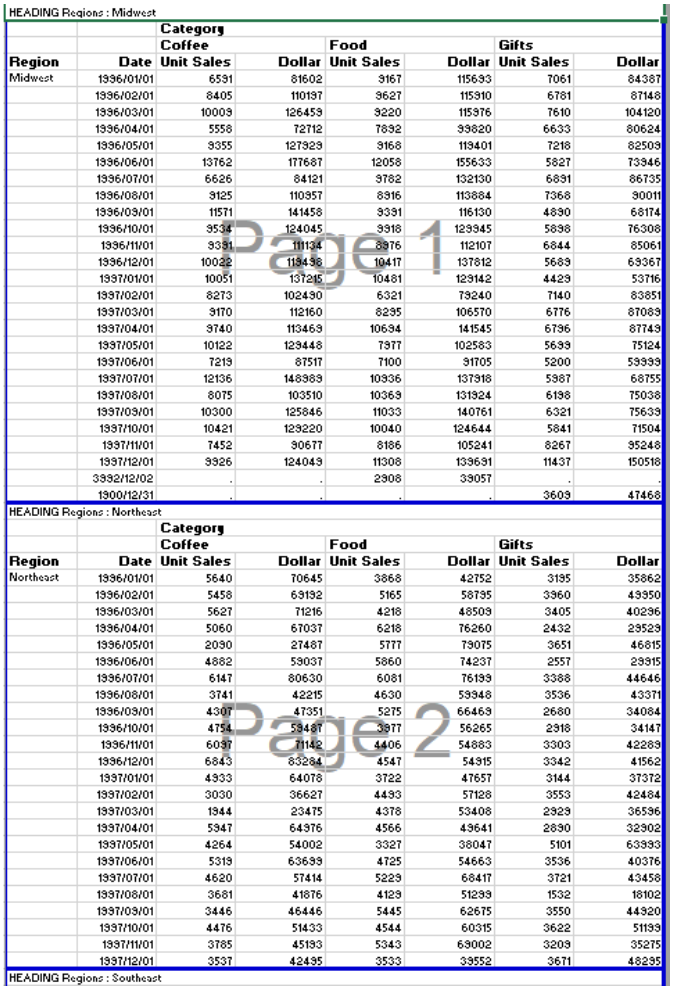
ON
: Does not synchronize FOCUS page breaks with Excel page breaks in format XLSX report output. This value conforms to behavior in prior releases.

Example: Synchronizing FOCUS Page Breaks With Excel Page Breaks in Format XLSX Report Output

The following request generates format XLSX report output with WebFOCUS page breaks that are inserted using the BY REGION PAGE-BREAK phrase.

```
SET XLSXPAGEBRKIGNORE=OFF
TABLE FILE GGSALES
"HEADING Regions : <REGION"
SUM UNITS DOLLARS
BY REGION PAGE-BREAK
BY DATE
ACROSS CATEGORY
WHERE CITY LE 'Memphis'
ON TABLE SET EXCELSERVURL LOCAL
ON TABLE HOLD AS XLSXPG1 FORMAT XLSX
ON TABLE SET STYLE *
XLSXPAGESETS=ON,
TOPMARGIN=1.25, BOTTOMMARGIN=1, ORIENTATION=LANDSCAPE, $
TYPE=REPORT, FONT=ARIAL, SIZE=9, $
TYPE=ACROSSTITLE, STYLE=BOLD, SIZE=10, $
TYPE=ACROSSVALUE, STYLE=BOLD, SIZE=10, $
TYPE=TITLE, STYLE=BOLD, SIZE=10, $
ENDSTYLE
END
```

Partial output is shown in the following image, using the Excel Page Break Preview view. XLSXPAGEBRKIGNORE is set to OFF (the default value). The default Excel page breaks are synchronized with the page breaks specified in the WebFOCUS request.



HEADING Regions : Midwest

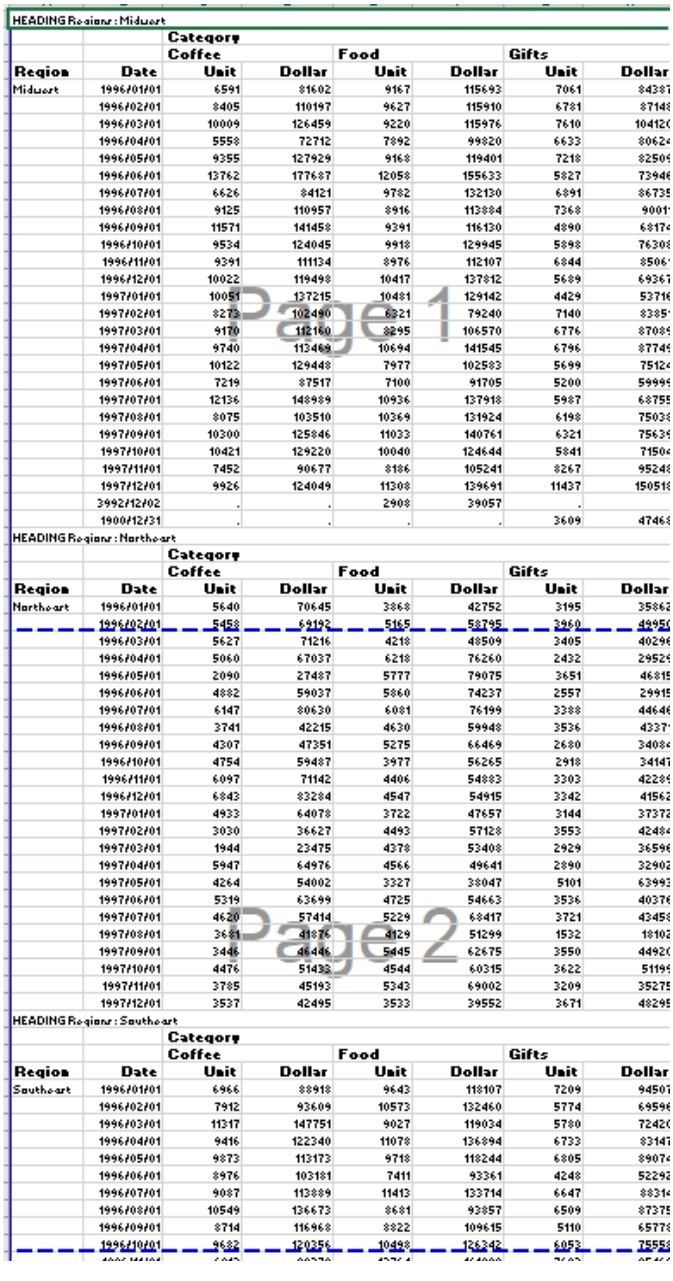
		Category					
		Coffee		Food		Gifts	
Region	Date	Unit Sales	Dollar	Unit Sales	Dollar	Unit Sales	Dollar
Midwest	1996/01/01	6531	81602	3167	115693	7061	84387
	1996/02/01	8405	110197	3627	115910	6781	87148
	1996/03/01	10009	126459	3220	115976	7610	104120
	1996/04/01	5558	72712	7892	99820	6633	80624
	1996/05/01	9355	127929	3168	119401	7218	82509
	1996/06/01	13762	177687	12058	155633	5827	73946
	1996/07/01	6626	84121	9782	132130	6891	86735
	1996/08/01	9125	110957	8316	113884	7368	90011
	1996/09/01	11571	141458	9331	116130	4890	68174
	1996/10/01	9534	124045	9318	123945	5898	76308
	1996/11/01	9381	111134	8876	112107	6844	85061
	1996/12/01	10022	113436	10417	137812	5689	69367
	1997/01/01	10051	137215	10481	123142	4429	53716
	1997/02/01	8273	102430	6321	79240	7140	83851
	1997/03/01	9170	112160	8295	106570	6776	87089
	1997/04/01	9740	113469	10694	141545	6796	87749
	1997/05/01	10122	123448	7977	102583	5699	75124
	1997/06/01	7219	87517	7100	91705	5200	59999
	1997/07/01	12136	148989	10936	137918	5987	68755
	1997/08/01	8075	103510	10369	131924	6198	75038
	1997/09/01	10300	125846	11033	140761	6321	75639
	1997/10/01	10421	129220	10040	124644	5841	71504
	1997/11/01	7452	90677	8186	105241	8267	95248
	1997/12/01	9926	124049	11308	139631	11437	150518
	3332/12/02	.	.	2908	33057	.	.
	1900/12/31	.	.	.	.	3609	47468

HEADING Regions : Northeast

		Category					
		Coffee		Food		Gifts	
Region	Date	Unit Sales	Dollar	Unit Sales	Dollar	Unit Sales	Dollar
Northeast	1996/01/01	5640	70645	3868	42752	3195	35862
	1996/02/01	5458	69192	5165	58795	3360	43950
	1996/03/01	5627	71216	4218	48509	3405	40296
	1996/04/01	5060	67037	6218	76260	2432	29529
	1996/05/01	2090	27487	5777	73075	3651	46815
	1996/06/01	4882	59037	5860	74237	2557	29915
	1996/07/01	6147	80630	6081	76199	3388	44646
	1996/08/01	3741	42215	4630	59348	3536	43371
	1996/09/01	4307	47351	5275	66469	2680	34084
	1996/10/01	4754	53487	3977	56265	2318	34147
	1996/11/01	6097	71142	4406	54883	3303	42289
	1996/12/01	6843	83284	4547	54315	3342	41562
	1997/01/01	4933	64078	3722	47657	3144	37372
	1997/02/01	3030	36627	4493	57128	3553	42484
	1997/03/01	1944	23475	4378	53408	2929	36596
	1997/04/01	5347	64376	4566	43641	2890	32902
	1997/05/01	4264	54002	3327	38047	5101	63993
	1997/06/01	5319	63693	4725	54663	3536	40376
	1997/07/01	4620	57414	5229	68417	3721	43458
	1997/08/01	3681	41876	4129	51299	1532	18102
	1997/09/01	3446	46446	5445	62675	3550	44920
	1997/10/01	4476	51433	4544	60315	3622	51199
	1997/11/01	3785	45193	5343	63002	3209	35275
	1997/12/01	3537	42495	3533	39552	3671	48295

HEADING Regions : Southeast

Changing the value of the XLSXPAGEBRKIGNORE parameter to ON produces the following partial output, in which the Excel page breaks are not synchronized with the FOCUS page breaks.



HEADING Region : Midwest

		Category					
		Coffee		Food		Gifts	
Region	Date	Unit	Dollar	Unit	Dollar	Unit	Dollar
Midwest	1996/01/01	6591	81602	9167	115693	7061	84387
	1996/02/01	8405	110197	9627	115910	6781	87148
	1996/03/01	10009	126459	9220	115976	7610	104120
	1996/04/01	5558	72712	7892	99820	6633	80624
	1996/05/01	9355	127929	9168	119401	7218	82506
	1996/06/01	13762	177687	12058	155633	5827	73946
	1996/07/01	6626	84121	9782	132130	6891	86735
	1996/08/01	9125	110957	8916	113884	7368	90017
	1996/09/01	11571	141458	9391	116130	4890	68174
	1996/10/01	9534	124045	9918	129945	5898	76308
	1996/11/01	9391	111134	8976	112107	6844	85067
	1996/12/01	10022	119498	10417	137812	5689	69267
	1997/01/01	10051	137215	10481	129142	4429	53716
	1997/02/01	8273	102490	8321	79240	7140	83857
	1997/03/01	9170	112160	8295	106570	6776	87085
	1997/04/01	9740	113469	10694	141545	6796	87745
	1997/05/01	10122	129448	7977	102583	5699	75124
	1997/06/01	7219	87517	7100	91705	5200	59999
	1997/07/01	12136	148989	10936	137918	5987	68755
	1997/08/01	8075	103510	10369	131924	6198	75038
	1997/09/01	10300	125846	11033	140761	6321	75639
	1997/10/01	10421	129220	10040	124644	5841	71504
	1997/11/01	7452	90677	8186	105241	8267	95248
	1997/12/01	9926	124049	11308	139691	11437	150518
	3992/12/02	.	.	2908	39057	.	
	1900/12/31	.	.	.	.	3609	47468

HEADING Region : Northeast

		Category					
		Coffee		Food		Gifts	
Region	Date	Unit	Dollar	Unit	Dollar	Unit	Dollar
Northeast	1996/01/01	5640	70645	3868	42752	3195	35862
	1996/02/01	5458	69192	5165	58795	3960	49950
	1996/03/01	5627	71216	4218	48509	3405	40296
	1996/04/01	5060	67037	6218	76260	2432	29525
	1996/05/01	2090	27487	5777	79075	3651	46815
	1996/06/01	4882	59037	5860	74237	2557	29915
	1996/07/01	6147	80630	6081	76199	3388	44646
	1996/08/01	3741	42215	4630	59948	3536	43371
	1996/09/01	4307	47351	5275	66469	2680	34084
	1996/10/01	4754	59487	3977	56265	2918	34141
	1996/11/01	6097	71142	4406	54883	3303	42289
	1996/12/01	6843	83284	4547	54915	3342	41562
	1997/01/01	4933	64078	3722	47657	3144	37372
	1997/02/01	3030	36627	4493	57128	3553	42484
	1997/03/01	1944	23475	4378	53408	2929	36596
	1997/04/01	5947	64976	4566	49641	2890	32902
	1997/05/01	4264	54002	3327	38047	5101	63993
	1997/06/01	5319	63699	4725	54663	3536	40376
	1997/07/01	4620	57414	5229	68417	3721	43458
	1997/08/01	3681	41876	4129	51299	1532	18102
	1997/09/01	3446	46446	5445	62675	3550	44920
	1997/10/01	4476	51433	4544	60315	3622	51199
	1997/11/01	3785	45193	5343	69002	3209	35275
	1997/12/01	3537	42495	3533	39552	3671	48295

HEADING Region : Southeast

		Category					
		Coffee		Food		Gifts	
Region	Date	Unit	Dollar	Unit	Dollar	Unit	Dollar
Southeast	1996/01/01	6966	88918	9643	118107	7209	94507
	1996/02/01	7912	93609	10573	132460	5774	69596
	1996/03/01	11317	147751	9027	119034	5780	72420
	1996/04/01	9416	122340	11078	136894	6733	83141
	1996/05/01	9873	113173	9718	118244	6805	89074
	1996/06/01	8976	103181	7411	93361	4248	52292
	1996/07/01	9087	113889	11413	133714	6647	88314
	1996/08/01	10549	136673	8681	93857	6509	87375
	1996/09/01	8714	116968	8822	109615	5110	65778
	1996/10/01	9682	120356	10498	126342	6053	75553

Scaling PDF Report Output to Fit the Page Width

By default, if PDF report output is too wide to fit on a single page, the report generates multiple panels of the same page for the columns that do not fit. The page numbers specify the page and panel numbers. For example, page numbers 1.1 and 1.2 represent page 1/panel 1 and page 1/panel 2.

You can scale the output to fit across the width of the page using the PAGE-SCALE StyleSheet attribute or the PAGE-SCALE SET parameter.

Syntax: How to Scale PDF Report Output to Fit the Page Width

In a StyleSheet, use the following syntax.

```
TYPE=REPORT, PAGE-SCALE={OFF|AUTO}, $
```

In a procedure or profile, use the following syntax.

```
SET PAGE-SCALE = {OFF|AUTO}
```

In a request, use the following syntax.

```
ON TABLE SET PAGE-SCALE {OFF|AUTO}
```

where:

`OFF`

Disables page scaling in PDF report output. This is the default value.

`AUTO`

Implements page scaling in PDF report output.

Reference: Usage Notes for PAGE-SCALE

- PAGE-SCALE is supported for PDF report output only.
- When a page is scaled to fit more content on the page horizontally, fewer vertical pages may be generated, as well.

Example: Scaling PDF Report Output to Fit the Page Width

The following request generates PDF report output without page scaling.

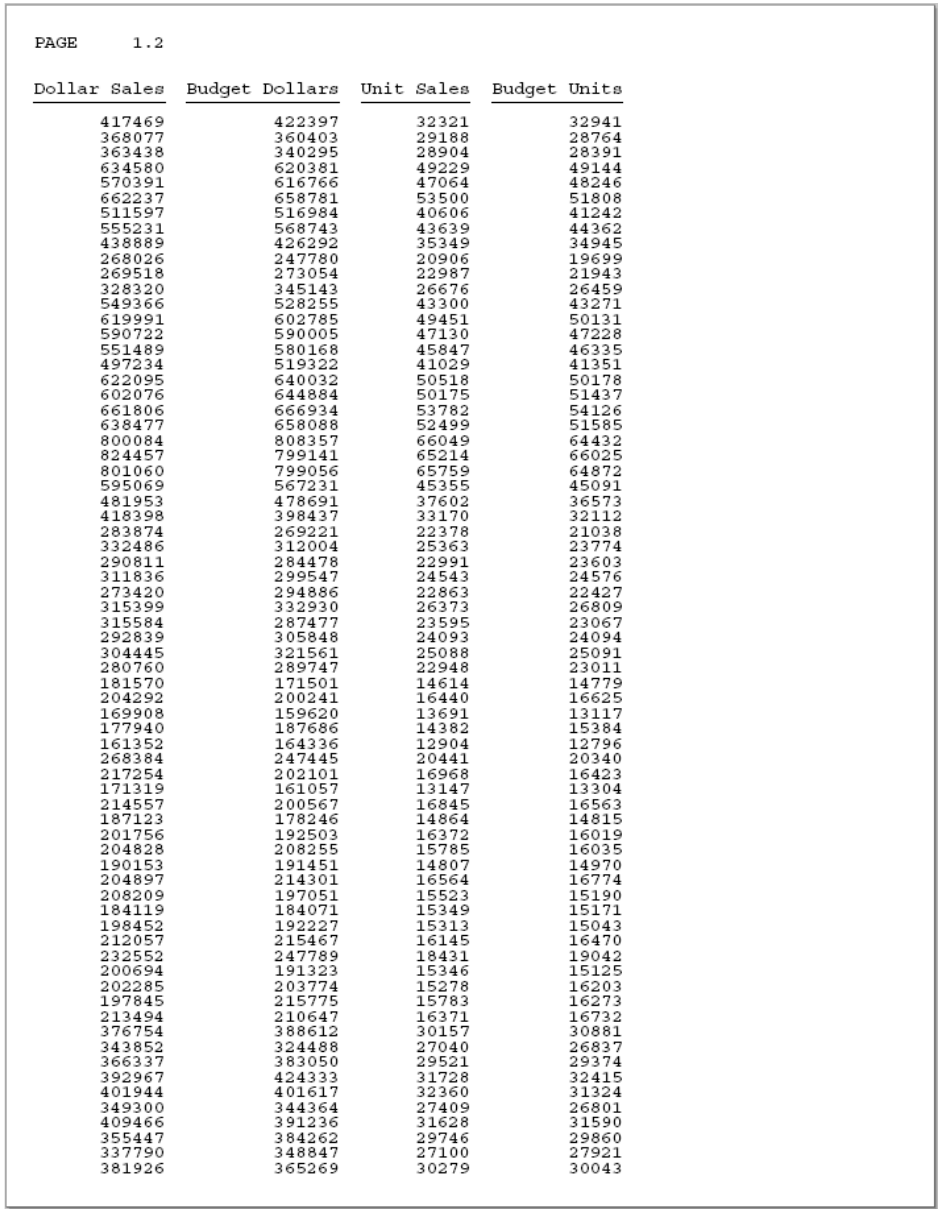
```
TABLE FILE GGSALES
SUM DOLLARS BUDDOLLARS UNITS BUDUNITS
BY CATEGORY
BY PRODUCT
BY REGION
BY ST
BY CITY
WHERE CATEGORY EQ 'Food' OR 'Gifts'
ON TABLE HOLD AS PGSCALE1 FORMAT PDF
END
```

The output is too wide for the page and is paneled. Page 1.1 has the columns that fit across the width of the page, as shown in the following image.

PAGE 1.1

Category	Product	Region	State	City
Food	Biscotti	Midwest	IL	Chicago
			MO	St. Louis
			TX	Houston
		Northeast	CT	New Haven
			MA	Boston
			NY	New York
		Southeast	FL	Orlando
			GA	Atlanta
			TN	Memphis
		West	CA	Los Angeles
				San Francisco
			WA	Seattle
	Croissant	Midwest	IL	Chicago
			MO	St. Louis
			TX	Houston
		Northeast	CT	New Haven
			MA	Boston
			NY	New York
		Southeast	FL	Orlando
			GA	Atlanta
			TN	Memphis
		West	CA	Los Angeles
				San Francisco
			WA	Seattle
	Scone	Midwest	IL	Chicago
			MO	St. Louis
			TX	Houston
		Northeast	CT	New Haven
			MA	Boston
			NY	New York
		Southeast	FL	Orlando
			GA	Atlanta
			TN	Memphis
		West	CA	Los Angeles
				San Francisco
			WA	Seattle
Gifts	Coffee Grinder	Midwest	IL	Chicago
			MO	St. Louis
			TX	Houston
		Northeast	CT	New Haven
			MA	Boston
			NY	New York
		Southeast	FL	Orlando
			GA	Atlanta
			TN	Memphis
		West	CA	Los Angeles
				San Francisco
			WA	Seattle
	Coffee Pot	Midwest	IL	Chicago
			MO	St. Louis
			TX	Houston
		Northeast	CT	New Haven
			MA	Boston
			NY	New York
		Southeast	FL	Orlando
			GA	Atlanta
			TN	Memphis
		West	CA	Los Angeles
				San Francisco
			WA	Seattle
	Mug	Midwest	IL	Chicago
			MO	St. Louis
			TX	Houston
		Northeast	CT	New Haven
			MA	Boston
			NY	New York
		Southeast	FL	Orlando
			GA	Atlanta
			TN	Memphis
		West	CA	Los Angeles

Page 1.2 has the remaining columns, as shown in the following image.



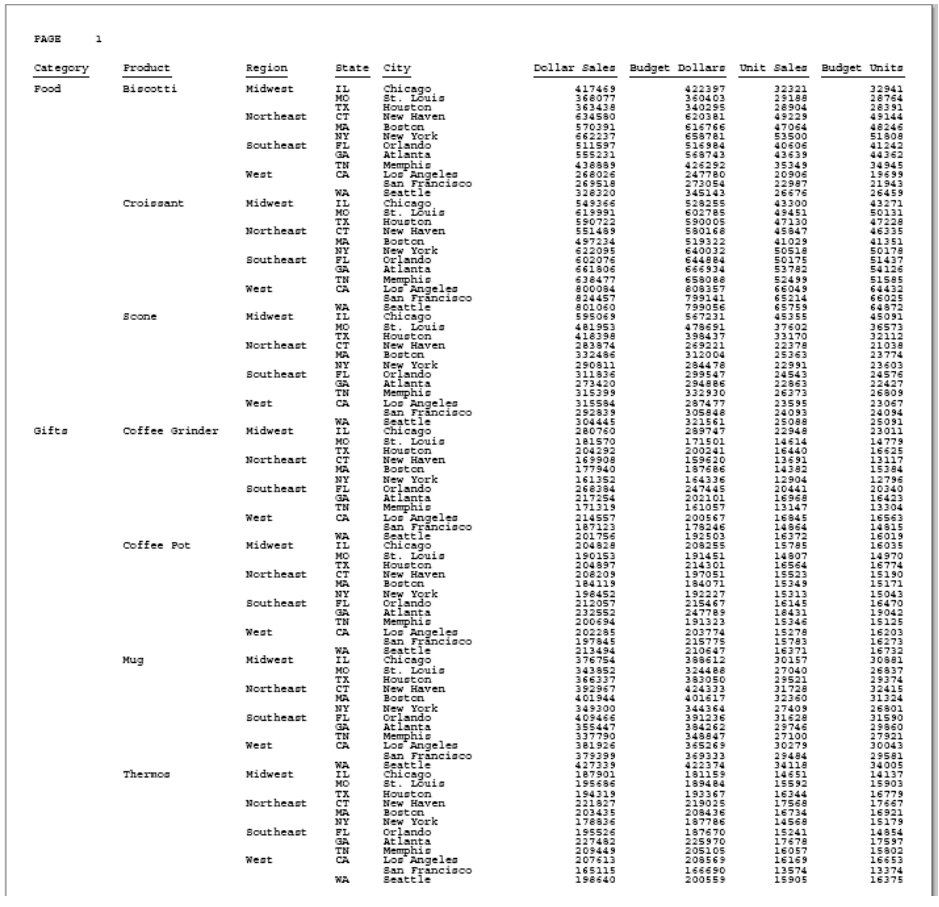
PAGE 1.2

Dollar Sales	Budget Dollars	Unit Sales	Budget Units
417469	422397	32321	32941
368077	360403	29188	28764
363438	340295	28904	28391
634580	620381	49229	49144
570391	616766	47064	48246
662237	658781	53500	51808
511597	516984	40606	41242
555231	568743	43639	44362
438889	426292	35349	34945
268026	247780	20906	19699
269518	273054	22987	21943
328320	345143	26676	26459
549366	528255	43300	43271
619991	602785	49451	50131
590722	590005	47130	47228
551489	580168	45847	46335
497234	519322	41029	41351
622095	640032	50518	50178
602076	644884	50175	51437
661806	666934	53782	54126
638477	658088	52499	51585
800084	808357	66049	64432
824457	799141	65214	66025
801060	799056	65759	64872
595069	567231	45355	45091
481953	478691	37602	36573
418398	398437	33170	32112
283874	269221	22378	21038
332486	312004	25363	23774
290811	284478	22991	23603
311836	299547	24543	24576
273420	294886	22863	22427
315399	332930	26373	26809
315584	287477	23595	23067
292839	305848	24093	24094
304445	321561	25088	25091
280760	289747	22948	23011
181570	171501	14614	14779
204292	200241	16440	16625
169908	159620	13691	13117
177940	187686	14382	15384
161352	164336	12904	12796
268384	247445	20441	20340
217254	202101	16968	16423
171319	161057	13147	13304
214557	200567	16845	16563
187123	178246	14864	14815
201756	192503	16372	16019
204828	208255	15785	16035
190153	191451	14807	14970
204897	214301	16564	16774
208209	197051	15523	15190
184119	184071	15349	15171
198452	192227	15313	15043
212057	215467	16145	16470
232552	247789	18431	19042
200694	191323	15346	15125
202285	203774	15278	16203
197845	215775	15783	16273
213494	210647	16371	16732
376754	388612	30157	30881
343852	324488	27040	26837
366337	383050	29521	29374
392967	424333	31728	32415
401944	401617	32360	31324
349300	344364	27409	26801
409466	391236	31628	31590
355447	384262	29746	29860
337790	348847	27100	27921
381926	365269	30279	30043

The following version of the request uses page scaling.

```
TABLE FILE GGSALES
SUM DOLLARS BUDDOLLARS UNITS BUDUNITS
BY CATEGORY
BY PRODUCT
BY REGION
BY ST
BY CITY
WHERE CATEGORY EQ 'Food' OR 'Gifts'
ON TABLE HOLD AS PGSCALE2 FORMAT PDF
ON TABLE SET STYLE *
TYPE=REPORT, PAGE-SCALE=AUTO,$
ENDSTYLE
END
```

The output is shown in the following image. All of the columns fit across the width of the page, with no paneling.



PAGE 1

Category	Product	Region	State	City	Dollar Sales	Budget Dollars	Unit Sales	Budget Units
Food	Biscotti	Midwest	IL	Chicago	417469	422387	32321	32941
			MO	St. Louis	368077	360403	29188	28764
			TX	Houston	363438	340295	28904	28391
		Northeast	CT	New Haven	634580	620381	49229	49144
			MA	Boston	570291	616786	47064	48845
			NY	New York	662237	659781	53900	51908
		Southeast	FL	Orlando	511597	516984	40606	41242
			GA	Atlanta	555231	568743	43539	44362
			TN	Memphis	438889	426292	35349	34545
		West	CA	Los Angeles	268026	247780	20906	19699
				San Francisco	269518	273054	22987	21943
			WA	Seattle	326320	345143	26676	26459
	Croissant	Midwest	IL	Chicago	549366	528255	43300	43271
			MO	St. Louis	619991	602785	49451	50131
			TX	Houston	590722	590005	47130	47228
		Northeast	CT	New Haven	551489	580168	45847	46335
			MA	Boston	497234	519322	41029	41351
			NY	New York	622095	640032	50518	50178
		Southeast	FL	Orlando	602076	644884	50175	51437
			GA	Atlanta	661806	666934	53782	54126
			TN	Memphis	536477	558088	52499	51585
		West	CA	Los Angeles	800084	808357	66049	64432
				San Francisco	824457	799141	68214	66025
			WA	Seattle	801060	799056	65759	64872
	Scone	Midwest	IL	Chicago	595069	567231	45355	45091
			MO	St. Louis	481953	478691	37602	36573
			TX	Houston	418398	398437	33170	32112
		Northeast	CT	New Haven	283874	269221	22378	21038
			MA	Boston	332486	312004	25363	23774
			NY	New York	290811	284478	22991	23603
		Southeast	FL	Orlando	311836	299547	24543	24576
			GA	Atlanta	273420	294886	22863	22427
			TN	Memphis	315399	332930	26273	26809
		West	CA	Los Angeles	315584	287477	23595	23087
				San Francisco	292839	305848	24093	24094
			WA	Seattle	304445	321561	25088	25091
Gifts	Coffee Grinder	Midwest	IL	Chicago	280760	289747	22948	23011
			MO	St. Louis	181570	171501	14614	14779
			TX	Houston	204292	200241	16440	16525
		Northeast	CT	New Haven	163908	159620	13691	13117
			MA	Boston	177940	187686	14382	15384
			NY	New York	161352	164336	12904	12796
		Southeast	FL	Orlando	268384	247445	20441	20340
			GA	Atlanta	217254	202101	16968	16423
			TN	Memphis	171318	161057	13147	13304
		West	CA	Los Angeles	214557	200567	16845	16563
				San Francisco	187123	178246	14864	14815
			WA	Seattle	201756	192503	16372	16019
	Coffee Pot	Midwest	IL	Chicago	204828	208285	15785	16035
			MO	St. Louis	190153	191451	14807	14970
			TX	Houston	204897	214301	16564	16774
		Northeast	CT	New Haven	208209	197081	16523	15190
			MA	Boston	184119	184071	15349	15171
			NY	New York	198452	192227	15313	15043
		Southeast	FL	Orlando	212057	215467	16145	16470
			GA	Atlanta	232852	247789	18431	19042
			TN	Memphis	200694	191323	15346	15125
		West	CA	Los Angeles	202285	203774	15278	16203
				San Francisco	197845	215775	15783	16273
			WA	Seattle	213494	210647	16371	16732
	Mug	Midwest	IL	Chicago	376754	388612	30157	30881
			MO	St. Louis	343852	324468	27040	26937
			TX	Houston	366337	383050	29521	29374
		Northeast	CT	New Haven	392967	424333	31728	32415
			MA	Boston	401944	401617	32360	31324
			NY	New York	349300	344364	27409	26801
		Southeast	FL	Orlando	409456	391236	31628	31590
			GA	Atlanta	355447	384262	28746	29860
			TN	Memphis	337790	348847	27100	27921
		West	CA	Los Angeles	381926	368269	30279	30043
				San Francisco	379299	369333	29484	29581
			WA	Seattle	427339	422374	34118	34005
	Thermos	Midwest	IL	Chicago	187901	181159	14651	14137
			MO	St. Louis	195686	189484	15592	15903
			TX	Houston	194319	193367	16344	16779
		Northeast	CT	New Haven	221827	219025	17568	17667
			MA	Boston	203435	208436	16734	16921
			NY	New York	176836	187786	14568	15179
		Southeast	FL	Orlando	195526	187670	15241	14854
			GA	Atlanta	227482	225970	17678	17587
			TN	Memphis	209449	205105	16057	15802
		West	CA	Los Angeles	207613	208569	16169	16653
				San Francisco	165115	166690	13574	13374
			WA	Seattle	198640	200559	15905	16375

Adapter Enhancements

This topic describes enhancements for data adapters.

Generating Table and Column Names With DBMS-Specific Length Limits

The name length limit usedin a Master File has been increased to the maximum length supported by the DBMS. In most cases this is 128 characters, although some have shorter limits, including the following:

- **Oracle.** 30 bytes.

- **DB2 on z/OS.** 30 bytes.

SQL Adapters: Optimizing OUTPUTLIMIT

An OUTPUTLIMIT filter in a TABLE request is now, by default, passed to some SQL data sources as FETCH FIRST n ROWS. When needed, it can be suppressed using the optimization setting FEATOPT OUTPUTLIMIT OFF.

For example, the following request against a Db2 data source contains an OUTPUTLIMIT filter:

```
SET TRACEUSER = ON
SET TRACEOFF = ALL
SET TRACEON = STMTRACE//CLIENT
TABLE FILE WFLITE
SUM COGS_US
BY PRODUCT_CATEGORY
IF OUTPUTLIMIT IS 50
END
SET TRACEUSER=OFF
```

The generated SQL request contains the FOR FETCH FIRST 50 ROWS clause:

```
SELECT
T1."ID_PRODUCT",
T1."COGS_US",
T7."ID_PRODUCT",
T7."PRODUCT_CATEGORY"
FROM
( wrd_wf_retail_sales T1
LEFT OUTER JOIN
wrd_wf_retail_product T7
ON T7."ID_PRODUCT" = T1."ID_PRODUCT" )
ORDER BY
T7."PRODUCT_CATEGORY"
FETCH FIRST 50 ROWS ONLY
FOR FETCH ONLY;
```

Raised Limits

The topic describes increases in limits.

Raised Limit for Join Fields

In prior releases, the limit for field pairs in a join was 20. The number of join field pairs has now been raised to 128.

Chapter 2

Upgrade Notes

This section describes changes in behaviour in this release.

In this chapter:

Software Branding

As of the following releases, ibi software and technical content are now branded under TIBCO® Software Inc.

- **WebFOCUS:** 8207.27.0
- **FOCUS:** 8207.27.0
- **iWay Service Manager:** 8.0.5
- **Omni-Gen:** 3.16.0

This change only impacts the names to which these products are referred. For example, WebFOCUS is now known as TIBCO WebFOCUS®, while iWay DataMigrator is now known as TIBCO® Data Migrator. You will begin to see this change throughout the software and corresponding technical content assets, including PDF covers and KnowledgeBase collections, where both new and former product names will be used interchangeably. For a full list of software branding, see below.

Former Product Name	New Product Name
WebFOCUS	TIBCO WebFOCUS®
WebFOCUS Client	TIBCO WebFOCUS® Client
WebFOCUS Server	TIBCO WebFOCUS® Reporting Server
iWay Data Migrator Server	TIBCO WebFOCUS® Reporting Server
iWay DataMigrator	TIBCO® Data Migrator

WebFOCUS App Studio	TIBCO WebFOCUS® App Studio
WebFOCUS Mobile App	TIBCO WebFOCUS® Mobile App
WebFOCUS Infographics	TIBCO WebFOCUS® Infographics
WebFOCUS Narrative Charts	TIBCO WebFOCUS® Narrative Charts
iWay Service Manager	TIBCO iWay® Service Manager
Omni-Gen	TIBCO Omni-Gen®
Omni Master Data Management	TIBCO Omni-Gen® MDM
Omni for Customer	TIBCO Omni-Gen® MDM
Omni-HealthData	TIBCO Omni-HealthData®
Omni-Insurance	TIBCO Omni-Insurance™
Data Quality	TIBCO Omni-Gen® Data Quality
FOCUS	TIBCO FOCUS®

Technical Content Branding

As of Release 8207.27.0, ibi products are now branded under TIBCO® Software Inc. This change only impacts the names to which these products are referred. Specifically, FOCUS is now known as TIBCO FOCUS®. You will begin to see this change throughout the software and corresponding technical content assets, including PDF covers and KnowledgeBase collections, where both new and former product names will be used interchangeably. For a full list of software branding, see *Software Branding* on page 57.

The following tables provide a quick overview of the PDF titles that have changed to support this release.

TIBCO FOCUS®

Former Title	**New Title**
FOCUS for Mainframe and Distributed Systems FOCUS Release Notes	*Release Notes*

FOCUS for Mainframe and Distributed Systems Overview and Operating Environments	*Overview and Operating Environments*
FOCUS for Mainframe and Distributed Systems Creating Reports	*Creating Reports*
FOCUS for Mainframe and Distributed Systems Describing Data	*Describing Data*
FOCUS for Mainframe and Distributed Systems Developing Applications	*Developing Applications*
FOCUS for Mainframe and Distributed Systems Maintaining Databases	*Maintaining Databases*
FOCUS for Mainframe and Distributed Systems Using Functions	*Using Functions*
FOCUS for Mainframe z/OS Installation Guide	*z/OS Installation Guide*
FOCUS for Mainframe Relational Data Adapter User's Manual	*Relational Data Adapter User's Manual*
FOCUS for Mainframe and Distributed Systems Adapter for Db2 Installation Guide	*Adapter for Db2 Installation Guide*
FOCUS for Mainframe Adapter for Teradata Installation Guide	*Adapter for Teradata Installation Guide*
FOCUS for Mainframe and Distributed Systems Active Technologies User Guide	*Active Technologies User Guide*
FOCUS for Mainframe Simultaneous Usage Reference Manual for z/OS	*Simultaneous Usage Reference Manual for z/OS*

Chapter 3

Known Issues

The following issues are known to occur in this release of TIBCO® FOCUS.

In this chapter:

Current Issues

- Pressing the Attention (ATTN) key is supposed to interrupt the current FOCUS process and bring up a menu of options that enables you to decide how to proceed. This is no longer happening. When you press Attention, you are taken to the READY prompt.

  Note: If you press *Enter* at this point you are taken back to where you were when you selected Attention.

SOME TIBCO SOFTWARE EMBEDS OR BUNDLES OTHER TIBCO SOFTWARE. USE OF SUCH EMBEDDED OR BUNDLED TIBCO SOFTWARE IS SOLELY TO ENABLE THE FUNCTIONALITY (OR PROVIDE LIMITED ADD-ON FUNCTIONALITY) OF THE LICENSED TIBCO SOFTWARE. THE EMBEDDED OR BUNDLED SOFTWARE IS NOT LICENSED TO BE USED OR ACCESSED BY ANY OTHER TIBCO SOFTWARE OR FOR ANY OTHER PURPOSE.

USE OF TIBCO SOFTWARE AND THIS DOCUMENT IS SUBJECT TO THE TERMS AND CONDITIONS OF A LICENSE AGREEMENT FOUND IN EITHER A SEPARATELY EXECUTED SOFTWARE LICENSE AGREEMENT, OR, IF THERE IS NO SUCH SEPARATE AGREEMENT, THE CLICKWRAP END USER LICENSE AGREEMENT WHICH IS DISPLAYED DURING DOWNLOAD OR INSTALLATION OF THE SOFTWARE (AND WHICH IS DUPLICATED IN THE LICENSE FILE) OR IF THERE IS NO SUCH SOFTWARE LICENSE AGREEMENT OR CLICKWRAP END USER LICENSE AGREEMENT, THE LICENSE(S) LOCATED IN THE "LICENSE" FILE(S) OF THE SOFTWARE. USE OF THIS DOCUMENT IS SUBJECT TO THOSE TERMS AND CONDITIONS, AND YOUR USE HEREOF SHALL CONSTITUTE ACCEPTANCE OF AND AN AGREEMENT TO BE BOUND BY THE SAME.

TIBCO, the TIBCO logo, the TIBCO O logo, FOCUS, iWay, Omni-Gen, Omni-HealthData, and WebFOCUS are either registered trademarks or trademarks of TIBCO Software Inc. in the United States and/or other countries.

Java and all Java based trademarks and logos are trademarks or registered trademarks of Oracle Corporation and/or its affiliates.

All other product and company names and marks mentioned in this document are the property of their respective owners and are mentioned for identification purposes only.

This software may be available on multiple operating systems. However, not all operating system platforms for a specific software version are released at the same time. See the readme file for the availability of this software version on a specific operating system platform.

THIS DOCUMENT IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT.

THIS DOCUMENT COULD INCLUDE TECHNICAL INACCURACIES OR TYPOGRAPHICAL ERRORS. CHANGES ARE PERIODICALLY ADDED TO THE INFORMATION HEREIN; THESE CHANGES WILL BE INCORPORATED IN NEW EDITIONS OF THIS DOCUMENT. TIBCO SOFTWARE INC. MAY MAKE IMPROVEMENTS AND/OR CHANGES IN THE PRODUCT(S) AND/OR THE PROGRAM(S) DESCRIBED IN THIS DOCUMENT AT ANY TIME.