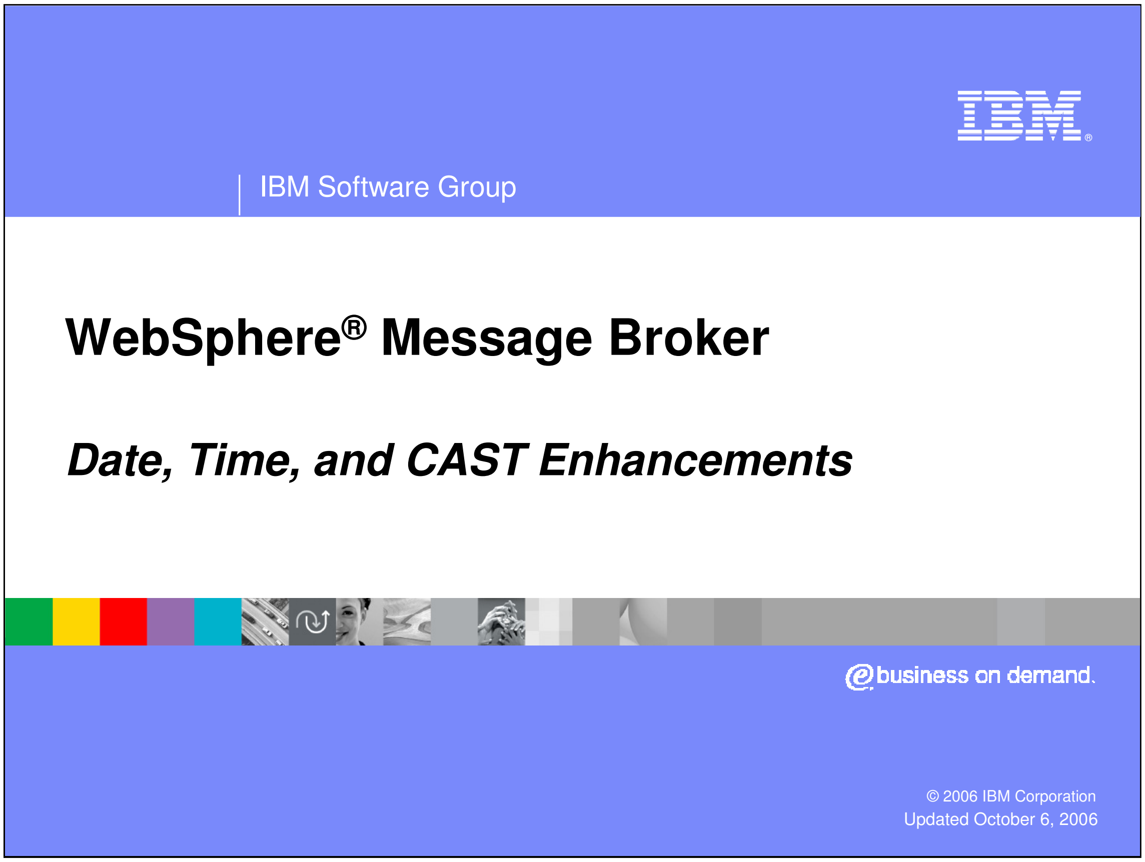


IBM Software Group

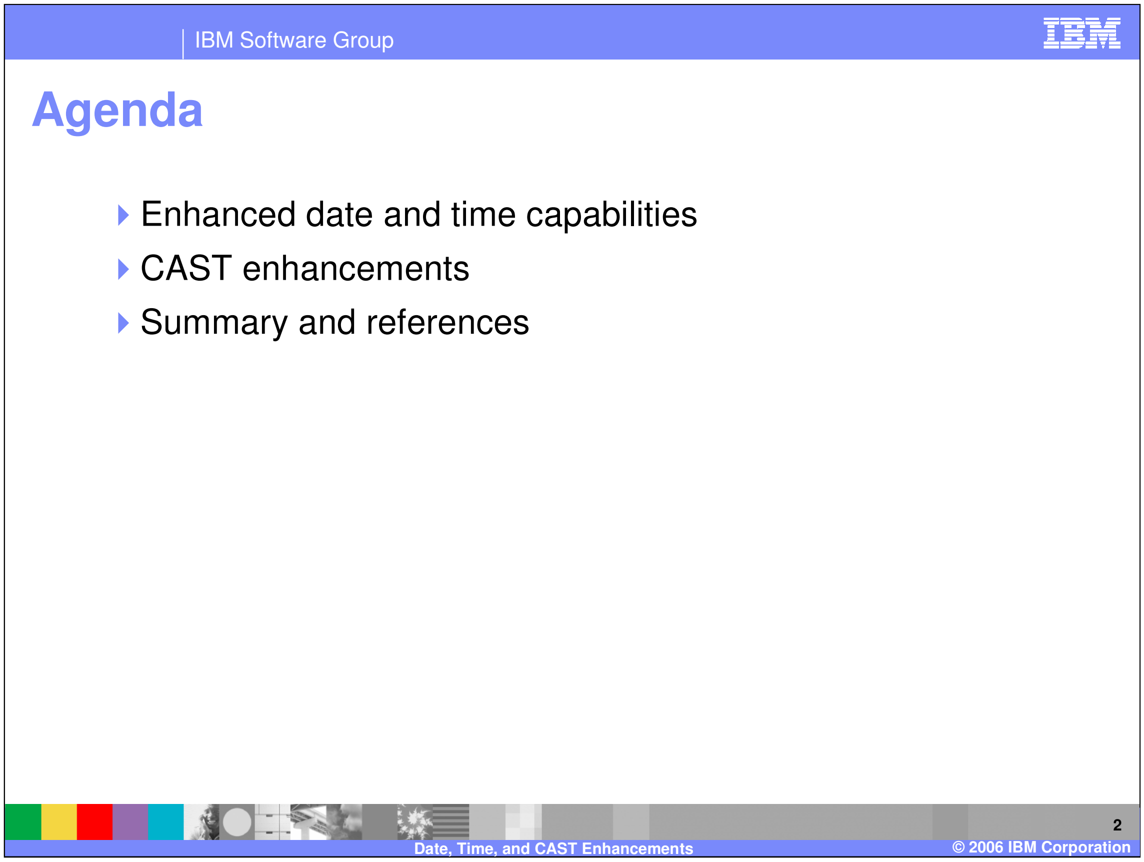
WebSphere® Message Broker

Date, Time, and CAST Enhancements

@business on demand.

Updated October 6, 2006

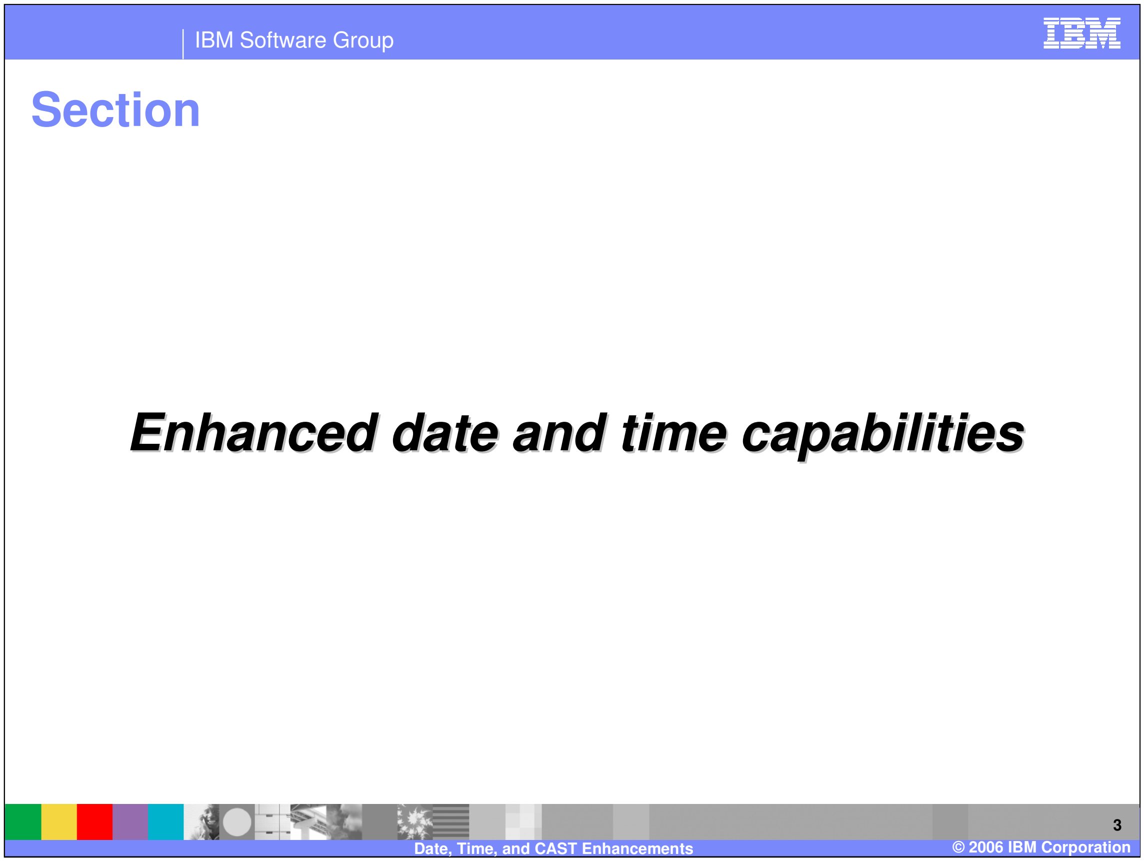
This presentation discusses the new enhancements for date and time ESQL functions and CAST functions in WebSphere Message Broker Version 6.



Agenda

- Enhanced date and time capabilities
- CAST enhancements
- Summary and references

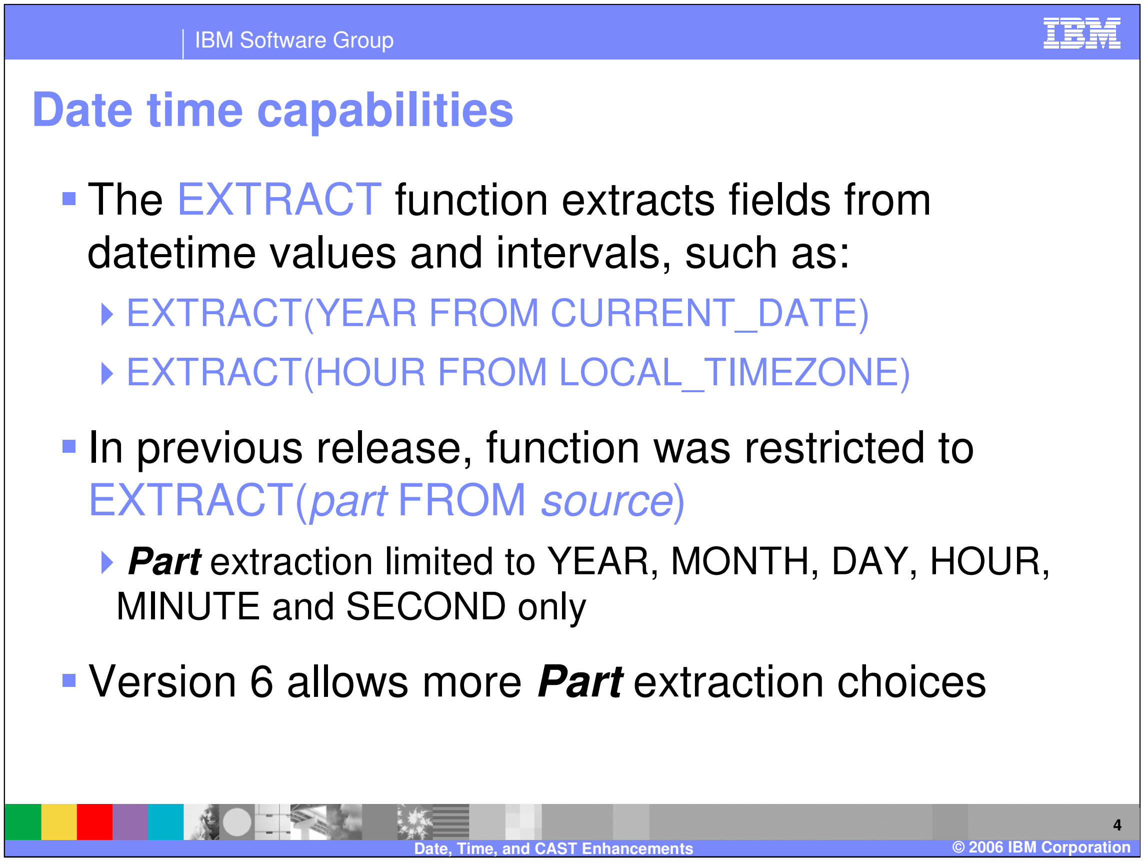
This presentation discusses Enhanced Date and Time Capabilities and CAST Enhancements. A Summary and References section concludes this presentation.



Section

Enhanced date and time capabilities

This section discusses the Enhanced Date and Time Capabilities in Version 6.

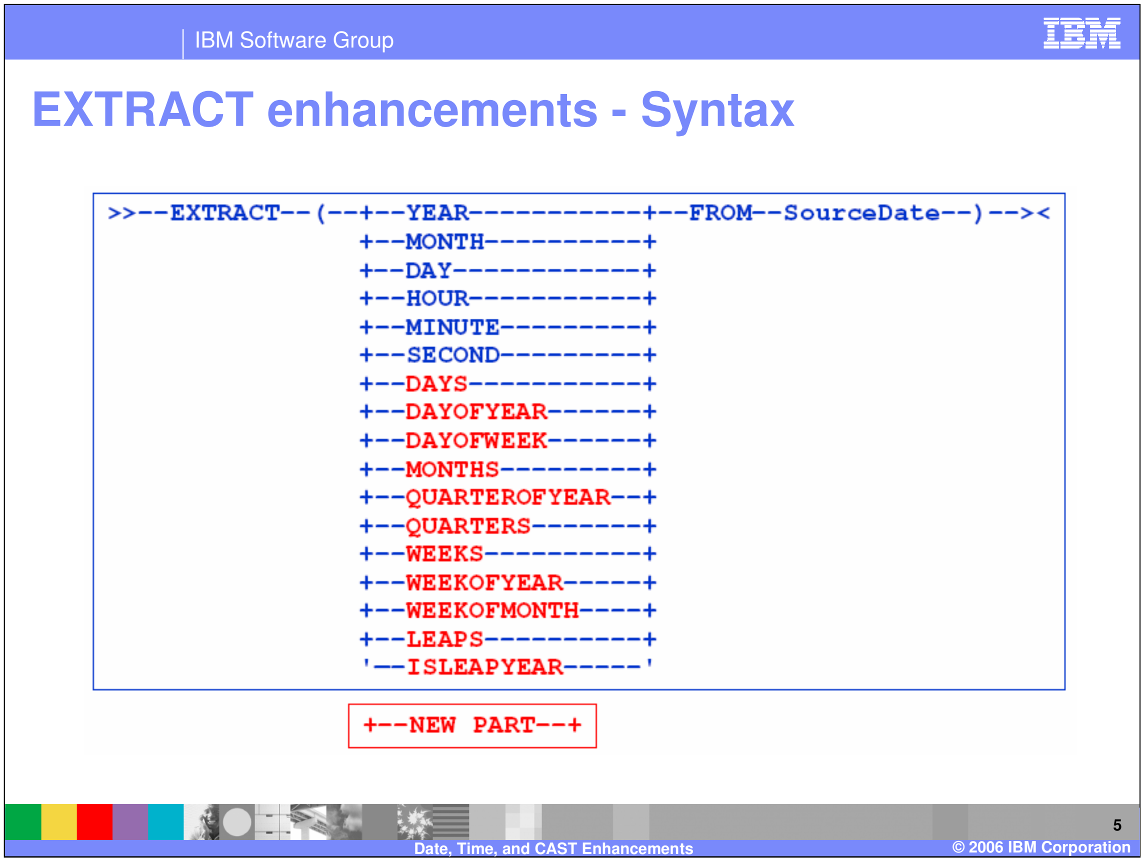


Date time capabilities

- The EXTRACT function extracts fields from datetime values and intervals, such as:
  - EXTRACT(YEAR FROM CURRENT_DATE)
  - EXTRACT(HOUR FROM LOCAL_TIMEZONE)
- In previous release, function was restricted to EXTRACT(*part* FROM *source*)
  - ***Part*** extraction limited to YEAR, MONTH, DAY, HOUR, MINUTE and SECOND only
- Version 6 allows more ***Part*** extraction choices

A significant date and time enhancement in Version 6 involves the EXTRACT function. The EXTRACT function extracts fields from “datetime” values and intervals, such as “EXTRACT YEAR FROM CURRENT_DATE”, or “EXTRACT HOUR FROM LOCAL_TIMEZONE”. In the previous release of WebSphere Message Broker, the “parts” of data that could be extracted was restricted to YEAR, MONTH, DAY, HOUR, MINUTE and SECOND.

In Version 6, WebSphere Message Broker allows more PART extraction choices with the EXTRACT function.

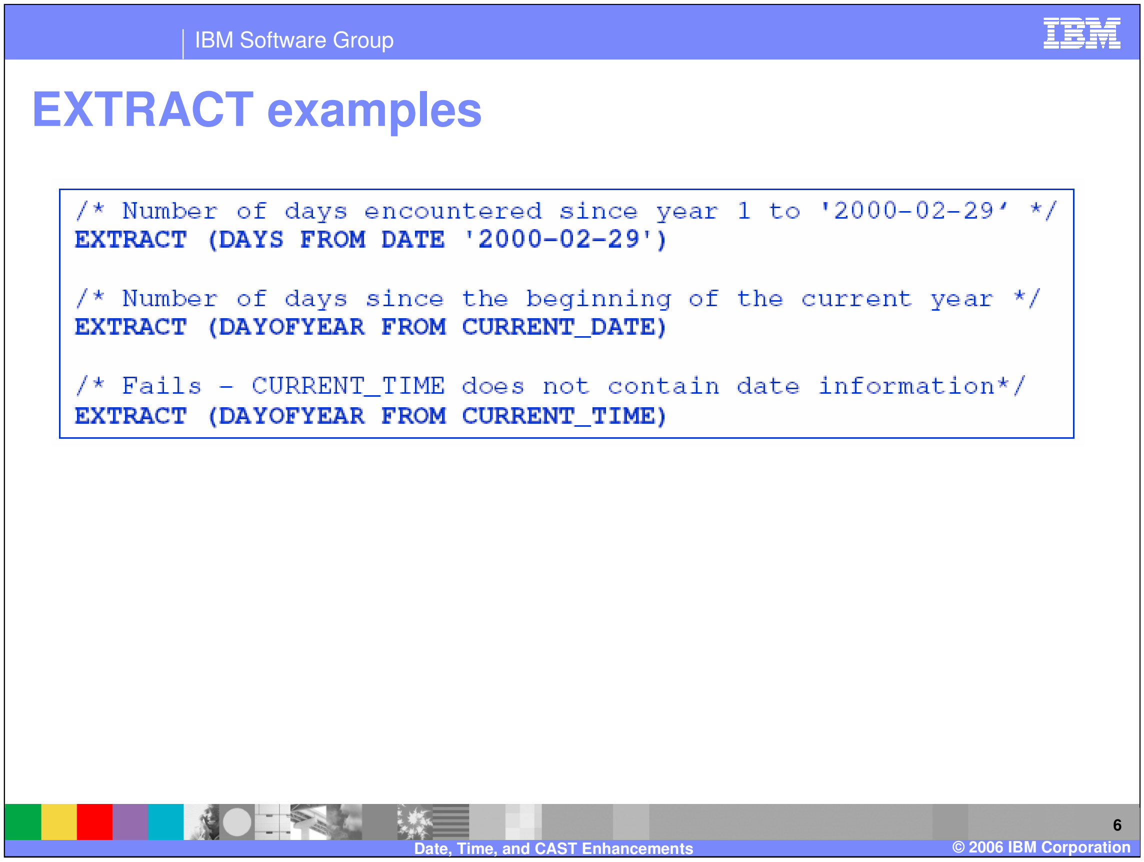


EXTRACT enhancements - Syntax

```
>>--EXTRACT--(--+--YEAR-----------+--FROM--SourceDate--)--><
                +--MONTH----------+
                +--DAY------------+
                +--HOUR-----------+
                +--MINUTE---------+
                +--SECOND---------+
                +--DAYS-----------+
                +--DAYOFYEAR------+
                +--DAYOFWEEK------+
                +--MONTHS---------+
                +--QUARTEROFYEAR--+
                +--QUARTERS-------+
                +--WEEKS----------+
                +--WEEKOFYEAR-----+
                +--WEEKOFMONTH----+
                +--LEAPS----------+
                '--ISLEAPYEAR-----'
```

+--NEW PART--+

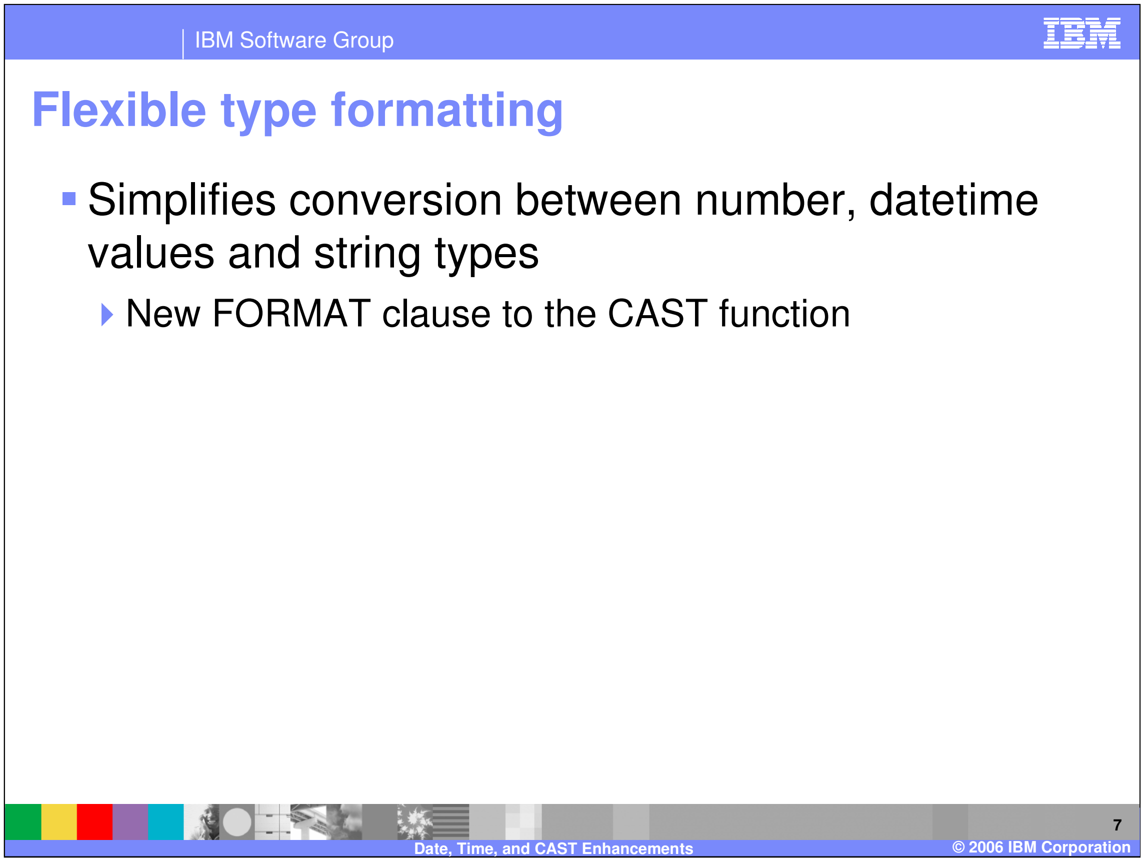
Here is the syntax diagram for the EXTRACT function. The 11 additional new choices for extraction of datetime values are shown in red. They are DAYS, DAYOFYEAR, DAYOFWEEK, MONTHS, QUARTEROFYEAR, QUARTERS, WEEKS, WEEKOFYEAR, WEEKOFMONTH, LEAPS, and ISLEAPYEAR. In the previous release, you were required to code the functionality provided by these choices. Now that they are provided as part of the product, they simplify your datetime value calculations and increase productivity and performance.



EXTRACT examples

```
/* Number of days encountered since year 1 to '2000-02-29‘ */
EXTRACT (DAYS FROM DATE '2000-02-29')

/* Number of days since the beginning of the current year */
EXTRACT (DAYOFYEAR FROM CURRENT_DATE)

/* Fails - CURRENT_TIME does not contain date information*/
EXTRACT (DAYOFYEAR FROM CURRENT_TIME)
```

Here are some examples using the EXTRACT function using “DAYS” and “DAYOFYEAR”. The Information Center provides additional information and examples.



Flexible type formatting

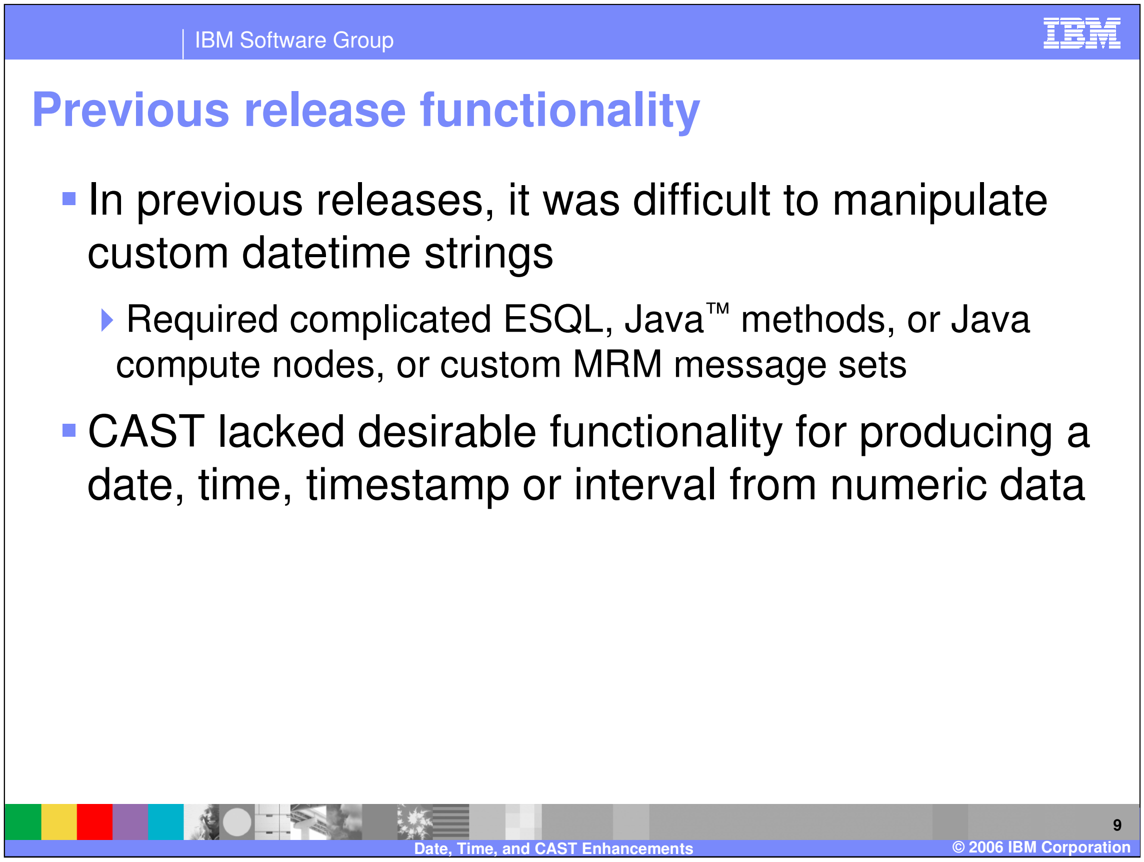
- Simplifies conversion between number, datetime values and string types
  - New FORMAT clause to the CAST function

In Version 6, the data type CASTing capabilities provide more flexibility when dealing with conversions between numeric values, datetime values, and string type values. Another significant enhancement to CAST which should be particularly helpful with datetime value processing is the new FORMAT clause. The CAST enhancements will be discussed in the next section of this presentation.

Section

CAST enhancements

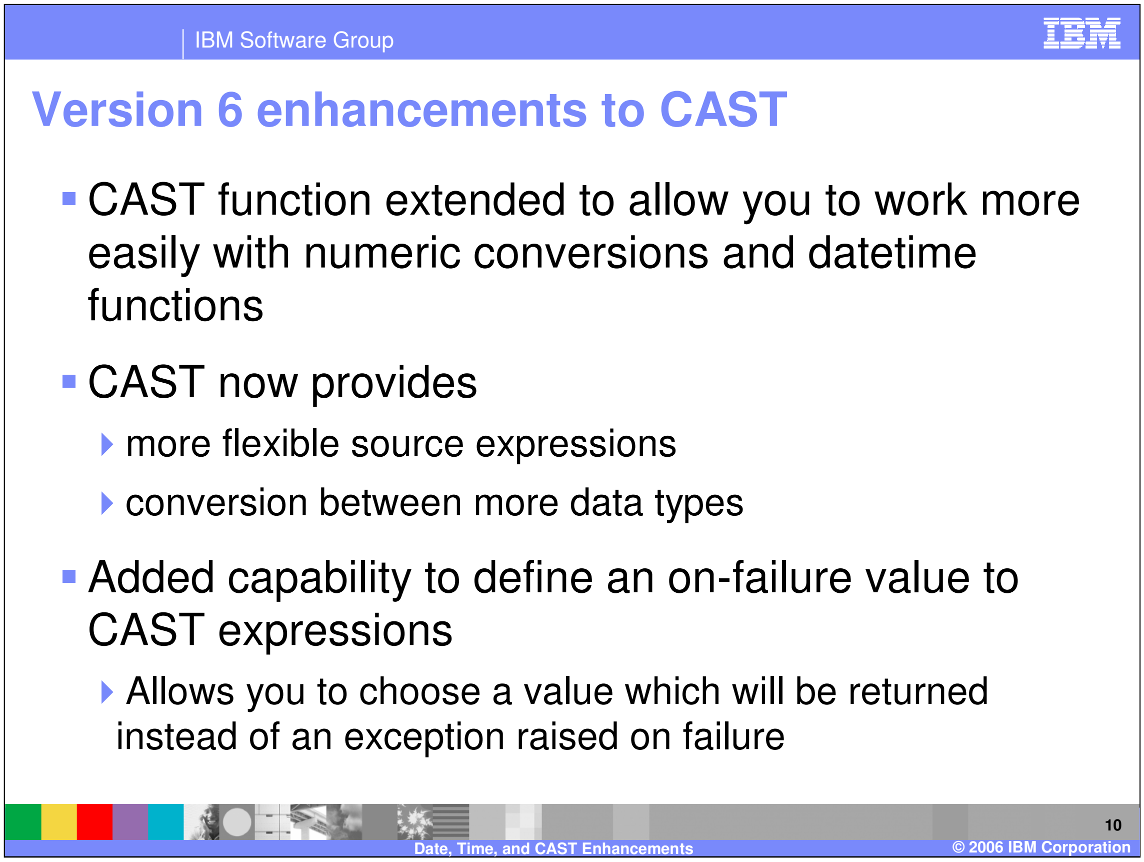
This section discusses CAST Enhancements in Version 6.



Previous release functionality

- In previous releases, it was difficult to manipulate custom datetime strings
  - Required complicated ESQL, Java™ methods, or Java compute nodes, or custom MRM message sets
- CAST lacked desirable functionality for producing a date, time, timestamp or interval from numeric data

In previous releases, it was difficult to manipulate custom date time strings. Often you had to code the functions you wanted in ESQL or in Java methods, or use Java compute nodes or custom MRM message sets.   In addition, CAST lacked functionality for producing a date, time, timestamp, or interval from numeric data.



IBM Software Group

IBM

Version 6 enhancements to CAST

- CAST function extended to allow you to work more easily with numeric conversions and datetime functions
- CAST now provides
  - more flexible source expressions
  - conversion between more data types
- Added capability to define an on-failure value to CAST expressions
  - Allows you to choose a value which will be returned instead of an exception raised on failure

10

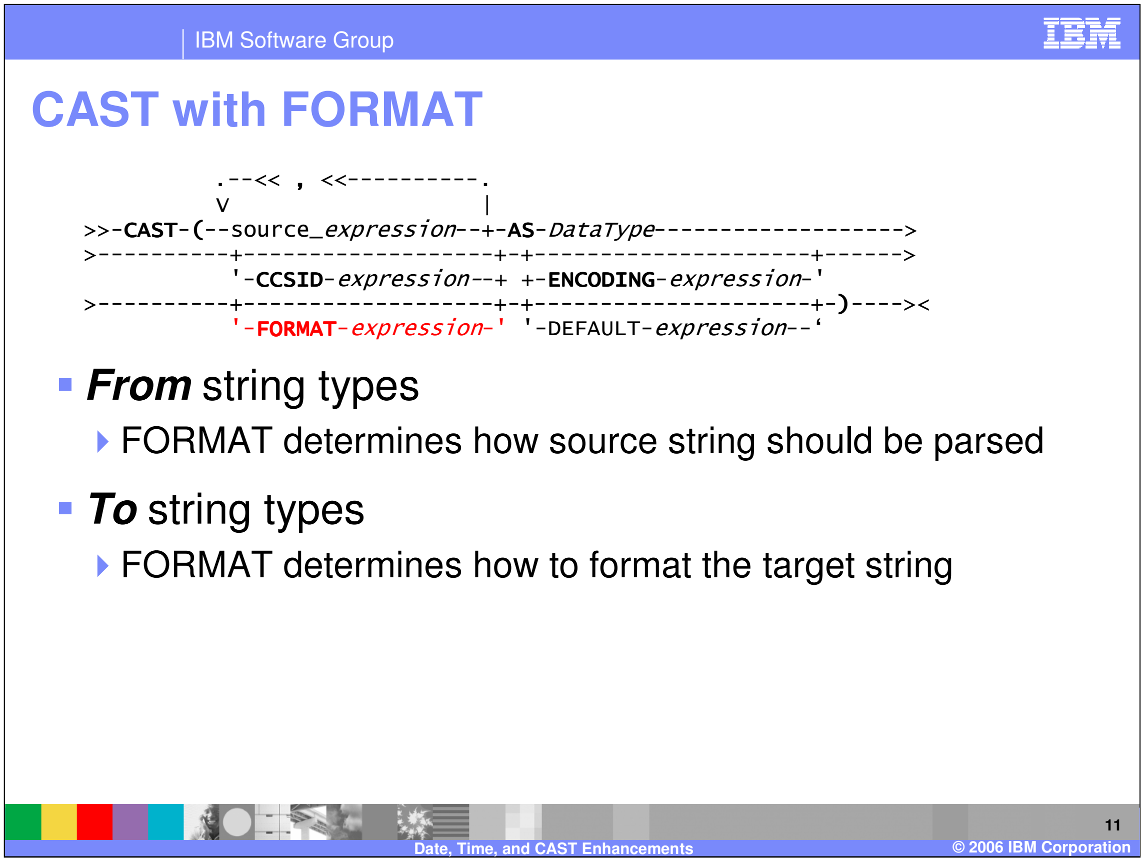
Date, Time, and CAST Enhancements

© 2006 IBM Corporation

The ESQL CAST function has been extended in Version 6 to allow you to work more easily with numeric conversions and datetime functions.

The CAST functionality is enhanced to support more flexible source expressions and to support conversion between more data types than was previously supported.

In addition, you can now define a user-defined “on-failure” value for ESQL CAST expressions, which allows you to chose a return value if a failure occurs, instead of an exception being raised.



IBM Software Group

CAST with FORMAT

```
          .--<< , <<----------.
          V                   |
>>-CAST-(--source_expression--+-AS-DataType------------------->
>----------+------------------+-+---------------------+------>
           '-CCSID-expression--+ +-ENCODING-expression-'
>----------+------------------+-+---------------------+-)---><
           '-FORMAT-expression-' '-DEFAULT-expression--‘
```

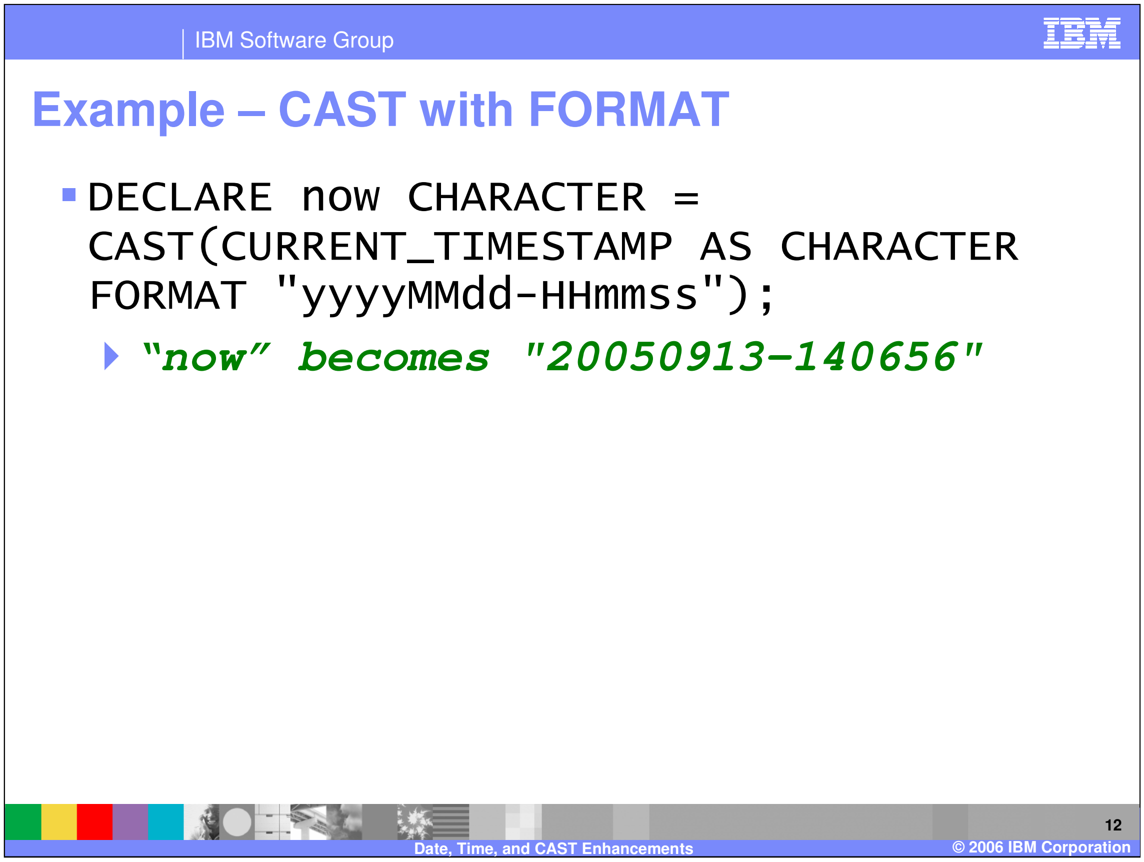
- ***From*** string types
  - FORMAT determines how source string should be parsed
- ***To*** string types
  - FORMAT determines how to format the target string

11

Date, Time, and CAST Enhancements

© 2006 IBM Corporation

A new FORMAT clause has been added to CAST in Version 6, shown here in this CAST syntax diagram. For conversions between string data types and numerical or datetime data types, you can supply this optional FORMAT clause. For conversions ***from*** string types, FORMAT defines how the source string should be parsed to fill the target data-type. For conversions ***to*** string types, it defines how the data in the source expression is to be formatted in the target string.



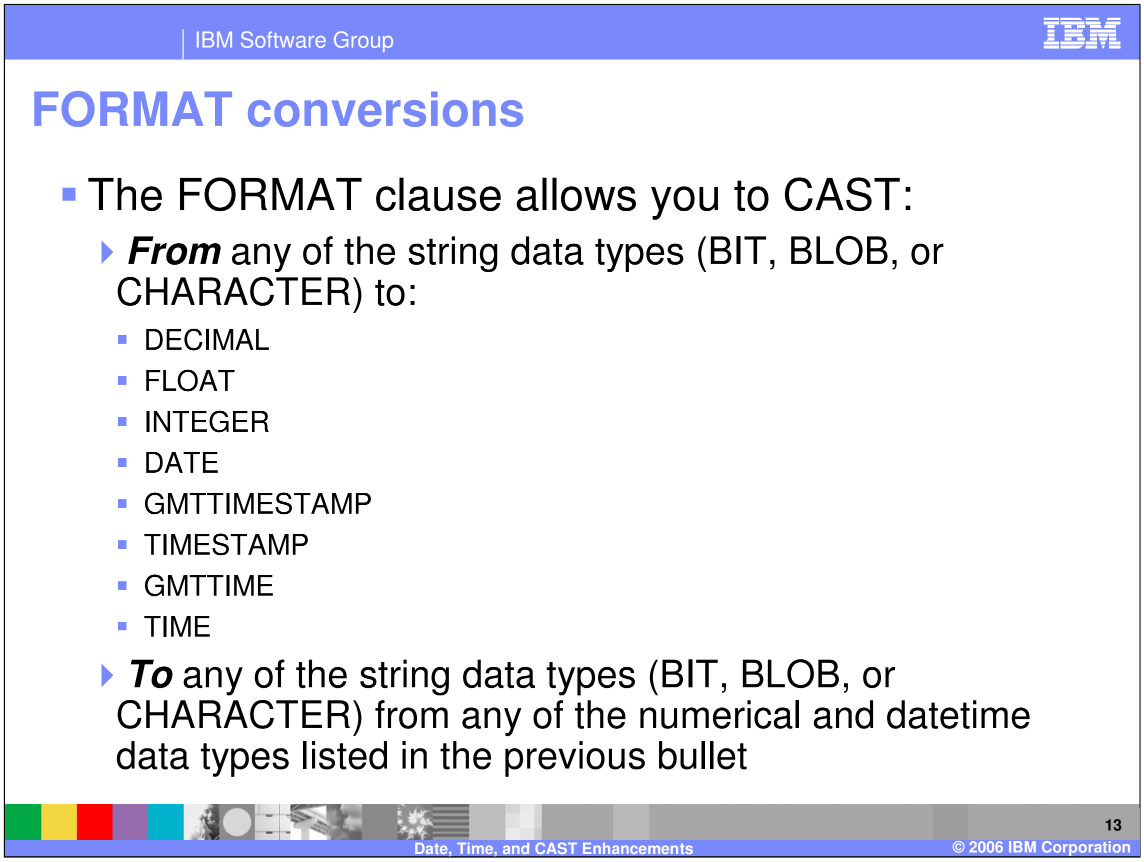
Example – CAST with FORMAT

```
DECLARE now CHARACTER =
CAST(CURRENT_TIMESTAMP AS CHARACTER
FORMAT "yyyyMMdd-HHmmss");
```

- *“now” becomes "20050913-140656"*

In this example, the target string variable called “now” with data type CHARACTER is created from CURRENT_TIMESTAMP, with a token string provided in quotes to indicate the required format for the target string variable.

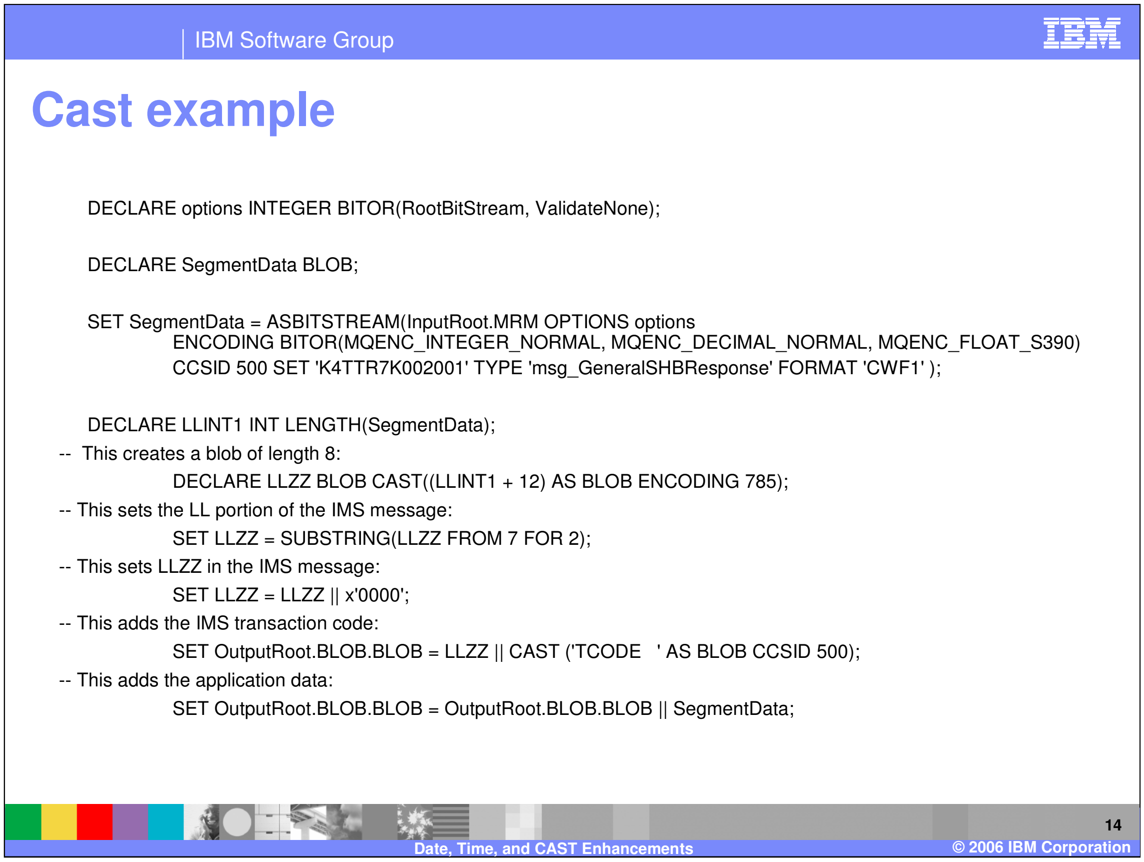
Review the WebSphere Message Broker Information Center to see more examples of CAST and the FORMAT clause.



FORMAT conversions

- The FORMAT clause allows you to CAST:
  - ***From*** any of the string data types (BIT, BLOB, or CHARACTER) to:
    - DECIMAL
    - FLOAT
    - INTEGER
    - DATE
    - GMTTIMESTAMP
    - TIMESTAMP
    - GMTTIME
    - TIME
  - ***To*** any of the string data types (BIT, BLOB, or CHARACTER) from any of the numerical and datetime data types listed in the previous bullet

The FORMAT clause allows you to CAST from data type BIT, BLOB or CHARACTER to the 8 data types listed above. Conversely, you can also CAST from those 8 data types to BIT, BLOB, or CHARACTER string data types.



IBM Software Group

Cast example

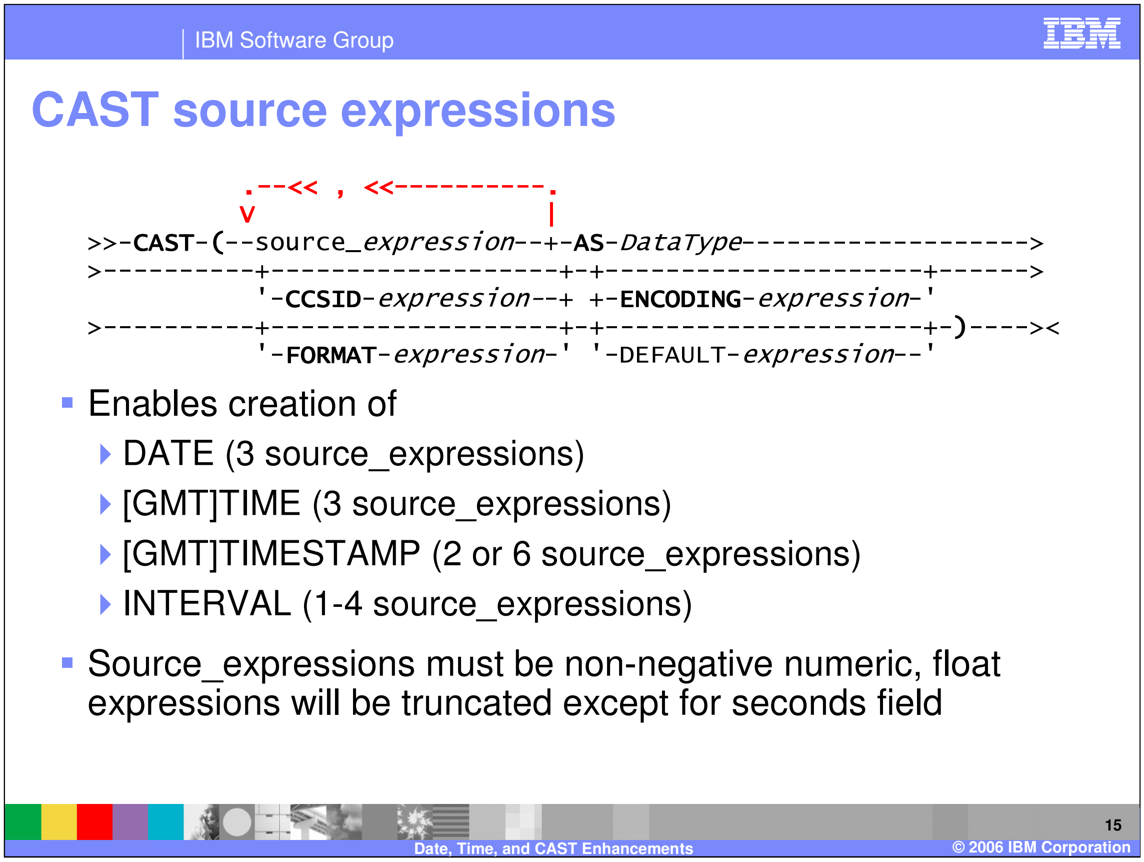
```
        DECLARE options INTEGER BITOR(RootBitStream, ValidateNone);

        DECLARE SegmentData BLOB;

        SET SegmentData = ASBITSTREAM(InputRoot.MRM OPTIONS options
                ENCODING BITOR(MQENC_INTEGER_NORMAL, MQENC_DECIMAL_NORMAL, MQENC_FLOAT_S390)
                CCSID 500 SET 'K4TTR7K002001' TYPE 'msg_GeneralSHBResponse' FORMAT 'CWF1' );

        DECLARE LLINT1 INT LENGTH(SegmentData);
-- This creates a blob of length 8:
                DECLARE LLZZ BLOB CAST((LLINT1 + 12) AS BLOB ENCODING 785);
-- This sets the LL portion of the IMS message:
                SET LLZZ = SUBSTRING(LLZZ FROM 7 FOR 2);
-- This sets LLZZ in the IMS message:
                SET LLZZ = LLZZ || x'0000';
-- This adds the IMS transaction code:
                SET OutputRoot.BLOB.BLOB = LLZZ || CAST ('TCODE   ' AS BLOB CCSID 500);
-- This adds the application data:
                SET OutputRoot.BLOB.BLOB = OutputRoot.BLOB.BLOB || SegmentData;
```

14

Date, Time, and CAST Enhancements

© 2006 IBM Corporation

This slide shows an example of building an IMS message on z/OS®.

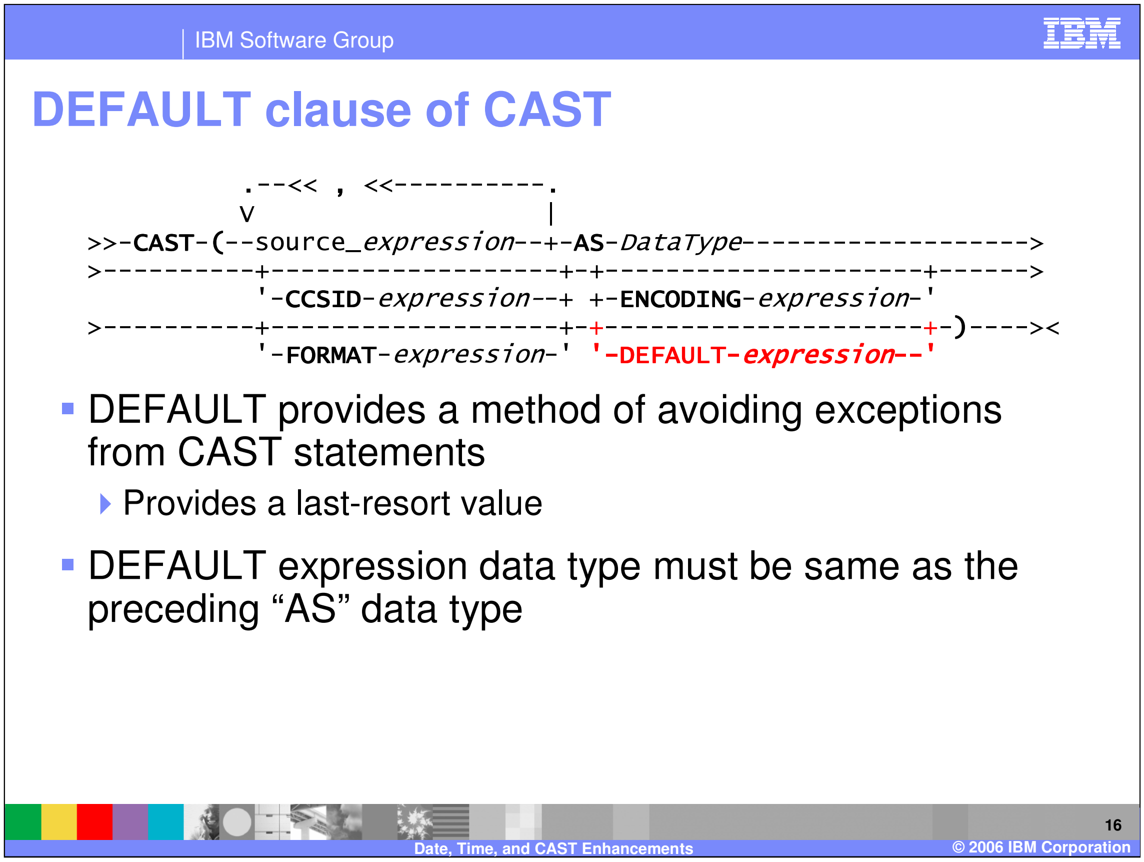


CAST source expressions

```
          .--<< , <<----------.
          v                   |
>>-CAST-(--source_expression--+-AS-DataType------------------->
>----------+-------------------+-+---------------------+------>
           '-CCSID-expression--+ +-ENCODING-expression-'
>----------+-------------------+-+---------------------+-)----><
           '-FORMAT-expression-' '-DEFAULT-expression--'
```

- Enables creation of
  - DATE (3 source_expressions)
  - [GMT]TIME (3 source_expressions)
  - [GMT]TIMESTAMP (2 or 6 source_expressions)
  - INTERVAL (1-4 source_expressions)
- Source_expressions must be non-negative numeric, float expressions will be truncated except for seconds field

CAST now permits more flexibility in source expressions than previous releases to allow you to produce a date, time, timestamp or interval from numeric data. Below the syntax diagram, you see a summary of the number of source expressions allowed for the various time expressions when performing this conversion. These source expressions must be non-negative, and “float” expressions used in these source expressions will be truncated, except for the “seconds” field.



IBM Software Group

IBM

DEFAULT clause of CAST

```
          .--<< , <<----------.
          V                   |
>>-CAST-(--source_expression--+-AS-DataType------------------>
>----------+------------------+-+---------------------+------>
           '-CCSID-expression--+ +-ENCODING-expression-'
>----------+------------------+-+---------------------+-)---><
           '-FORMAT-expression-' '-DEFAULT-expression--'
```

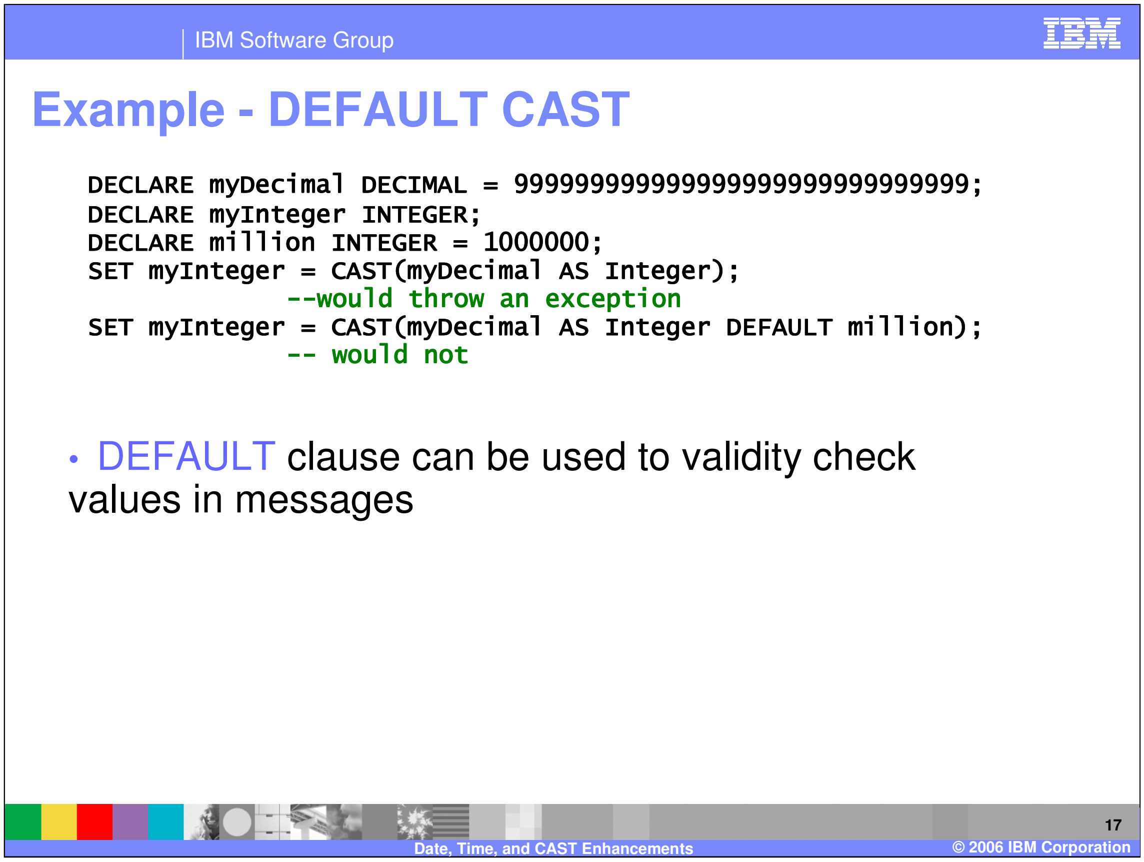
- DEFAULT provides a method of avoiding exceptions from CAST statements
  - Provides a last-resort value
- DEFAULT expression data type must be same as the preceding “AS” data type

16

Date, Time, and CAST Enhancements

© 2006 IBM Corporation

The new DEFAULT clause can be used with CAST to provide a last-resort value setting in the event that CASTing the source expression would cause an exception. The data type of the DEFAULT expression must be the same as the preceding “AS” data type; otherwise an exception will be thrown.



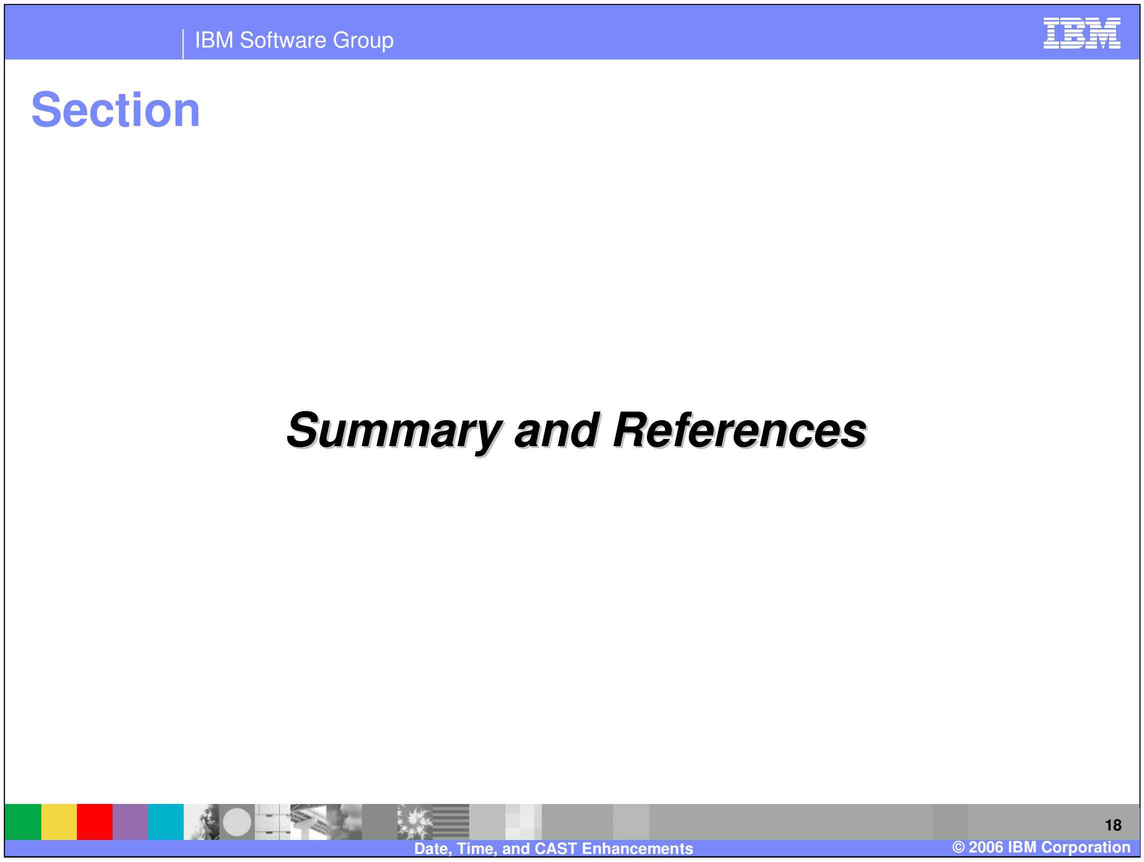
Example - DEFAULT CAST

```
DECLARE myDecimal DECIMAL = 999999999999999999999999999999;
DECLARE myInteger INTEGER;
DECLARE million INTEGER = 1000000;
SET myInteger = CAST(myDecimal AS Integer);
             --would throw an exception
SET myInteger = CAST(myDecimal AS Integer DEFAULT million);
             -- would not
```

- DEFAULT clause can be used to validity check values in messages

Here is an example of how DEFAULT can be used to prevent an exception. The first “SET” statement throws an exception because the large Decimal value cannot be cast as an Integer. In the second “SET” statement, the DEFAULT statement provides a return value for the CAST, preventing the exception.

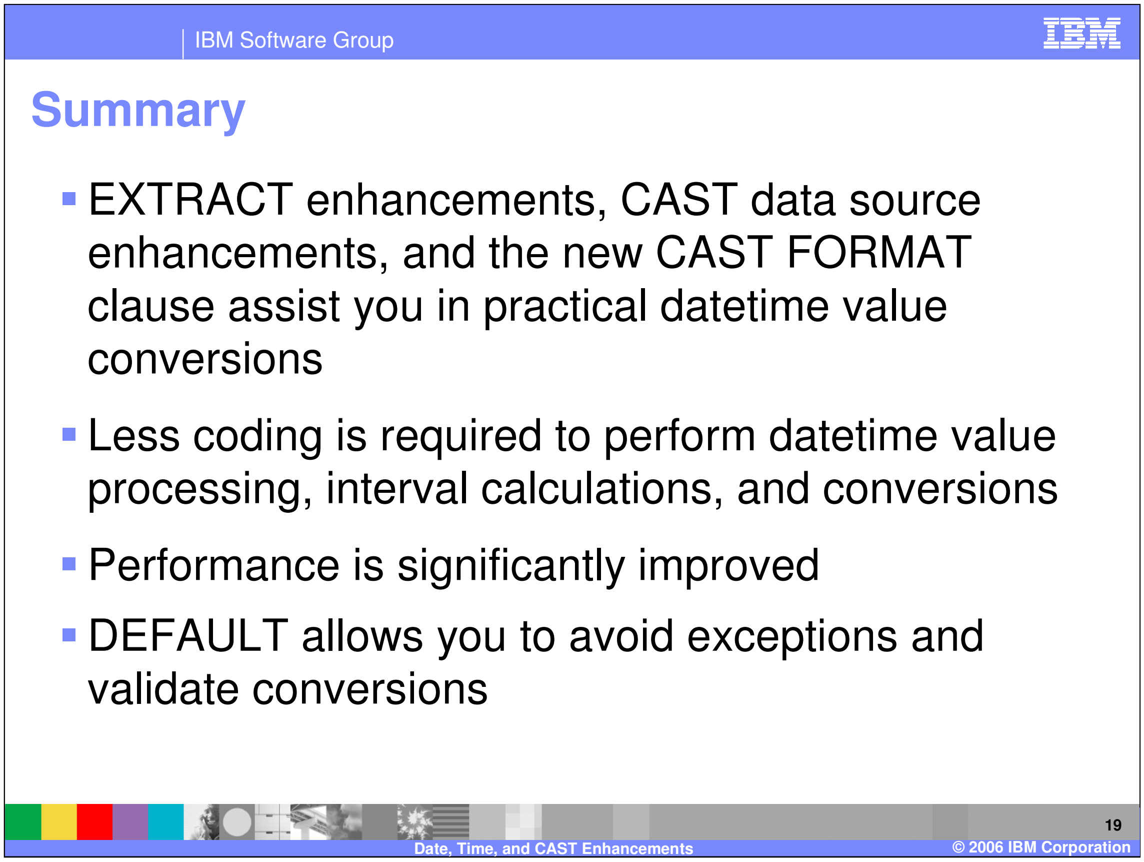
Because the value you specify for the DEFAULT is predictable, you can use the CAST function with the DEFAULT clause as a technique to validity check values in messages. If your DEFAULT value is returned, then you know the source cannot be CAST, and thus you can change the processing for that message field without incurring or handling an exception.



Section

Summary and References

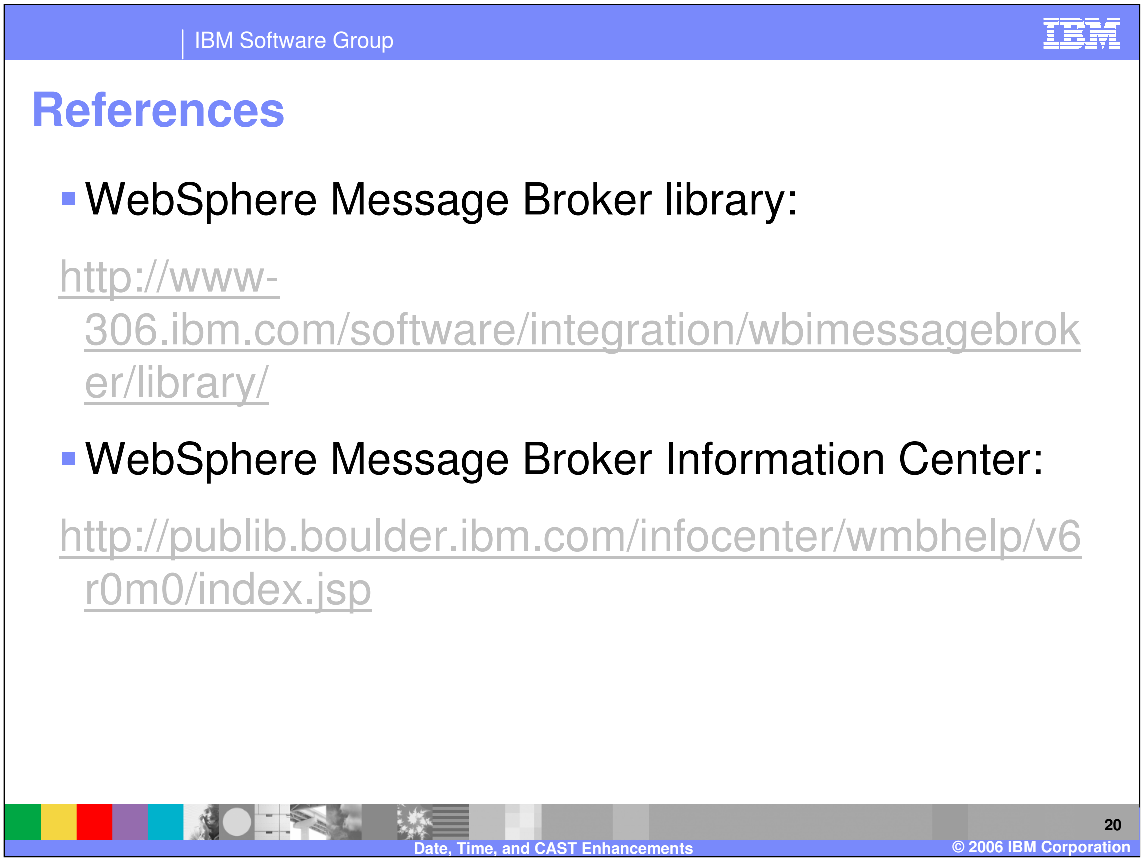
The last portion of the presentation contains a summary and references.



Summary

- EXTRACT enhancements, CAST data source enhancements, and the new CAST FORMAT clause assist you in practical datetime value conversions
- Less coding is required to perform datetime value processing, interval calculations, and conversions
- Performance is significantly improved
- DEFAULT allows you to avoid exceptions and validate conversions

The EXTRACT and CAST enhancements help you work with datetime values and numeric conversions in very efficient and practical ways. They are based on real-world requirements. These enhancements provide more flexibility in programming and remove the need to provide this functionality in your own ESQL code, which simplifies your ESQL code and improves performance. The new DEFAULT clause of CAST provides you a way to prevent exceptions and to validate your data at the same time.



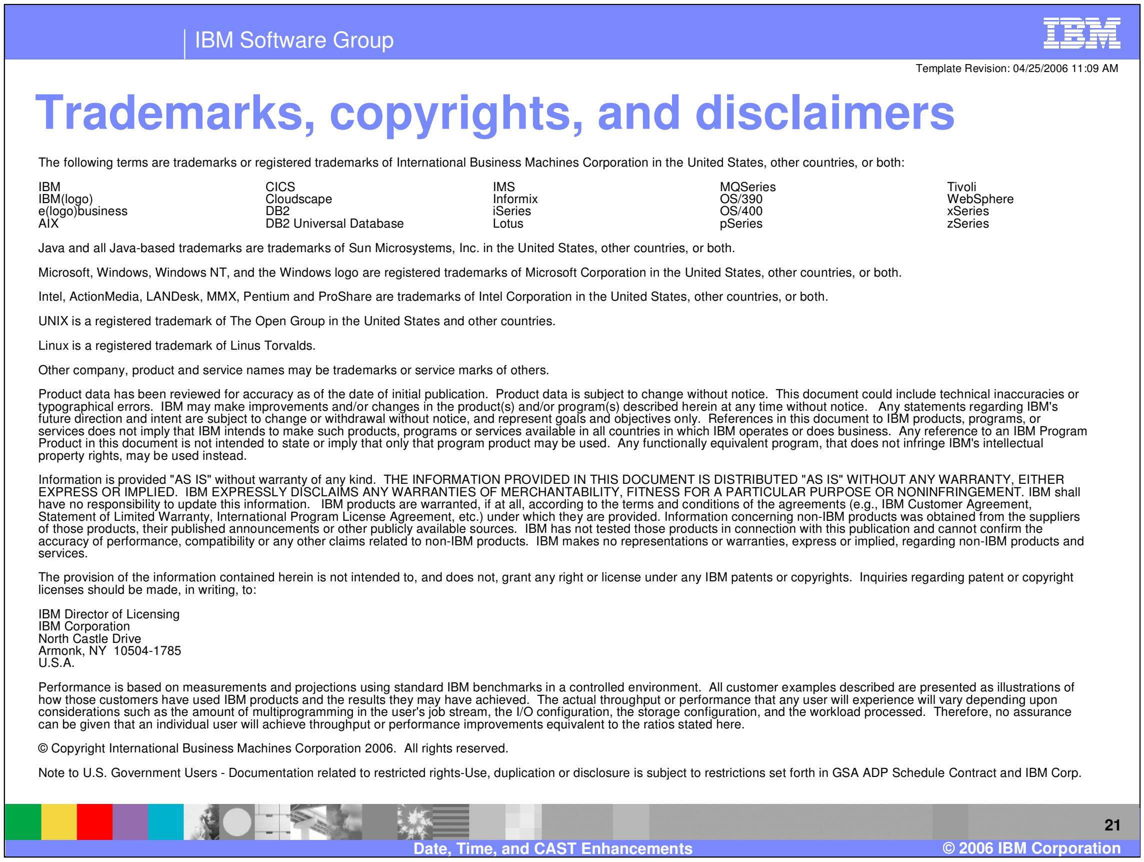
References

- WebSphere Message Broker library:

http://www-306.ibm.com/software/integration/wbimessagebroker/library/

- WebSphere Message Broker Information Center:

http://publib.boulder.ibm.com/infocenter/wmbhelp/v6r0m0/index.jsp



Template Revision: 04/25/2006 11:09 AM

Trademarks, copyrights, and disclaimers

The following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both:

IBM
IBM(logo)
e(logo)business
AIX
CICS
Cloudscape
DB2
DB2 Universal Database
IMS
Informix
iSeries
Lotus
MQSeries
OS/390
OS/400
pSeries
Tivoli
WebSphere
xSeries
zSeries

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are registered trademarks of Microsoft Corporation in the United States, other countries, or both.

Intel, ActionMedia, LANDesk, MMX, Pentium and ProShare are trademarks of Intel Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Linux is a registered trademark of Linus Torvalds.

Other company, product and service names may be trademarks or service marks of others.

Product data has been reviewed for accuracy as of the date of initial publication. Product data is subject to change without notice. This document could include technical inaccuracies or typographical errors. IBM may make improvements and/or changes in the product(s) and/or program(s) described herein at any time without notice. Any statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only. References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business. Any reference to an IBM Program Product in this document is not intended to state or imply that only that program product may be used. Any functionally equivalent program, that does not infringe IBM's intellectual property rights, may be used instead.

Information is provided "AS IS" without warranty of any kind. THE INFORMATION PROVIDED IN THIS DOCUMENT IS DISTRIBUTED "AS IS" WITHOUT ANY WARRANTY, EITHER EXPRESS OR IMPLIED. IBM EXPRESSLY DISCLAIMS ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NONINFRINGEMENT. IBM shall have no responsibility to update this information. IBM products are warranted, if at all, according to the terms and conditions of the agreements (e.g., IBM Customer Agreement, Statement of Limited Warranty, International Program License Agreement, etc.) under which they are provided. Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products in connection with this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. IBM makes no representations or warranties, express or implied, regarding non-IBM products and services.

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents or copyrights. Inquiries regarding patent or copyright licenses should be made, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. The actual throughput or performance that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput or performance improvements equivalent to the ratios stated here.

Note to U.S. Government Users - Documentation related to restricted rights-Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract and IBM Corp.