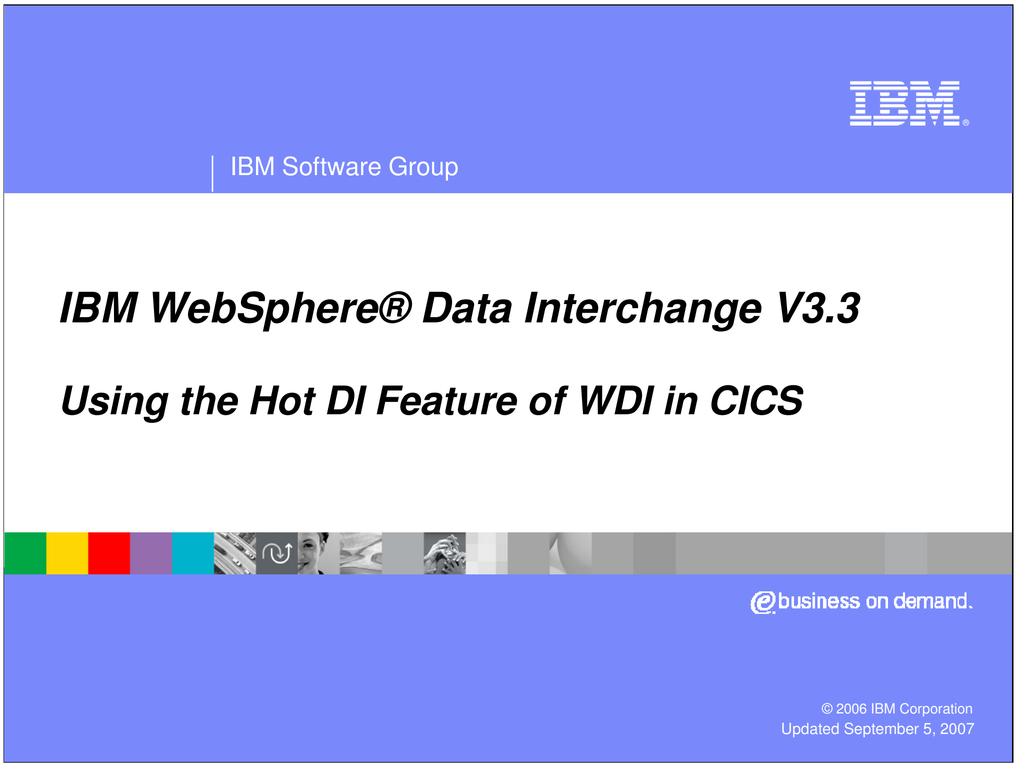


IBM Software Group

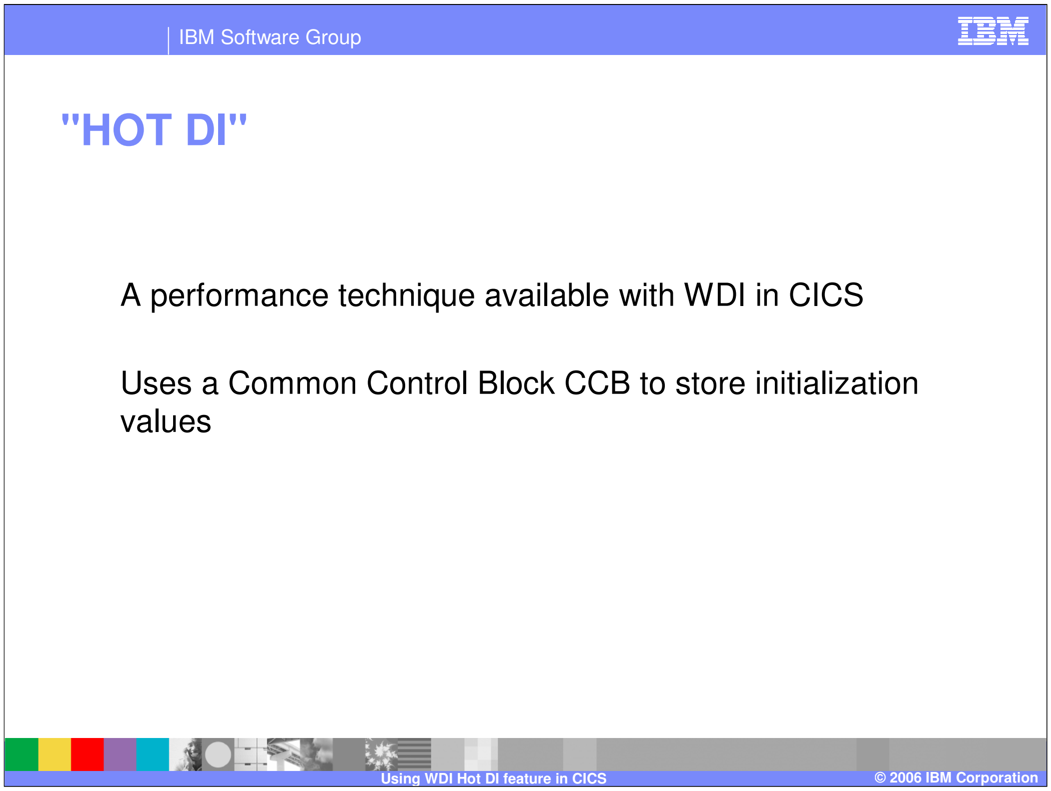
IBM WebSphere® Data Interchange V3.3

Using the Hot DI Feature of WDI in CICS

business on demand.

Updated September 5, 2007

This presentation will assist the user in understanding the functioning of the Hot DI feature of WDI and how it can help translation performance in the CICS environment.

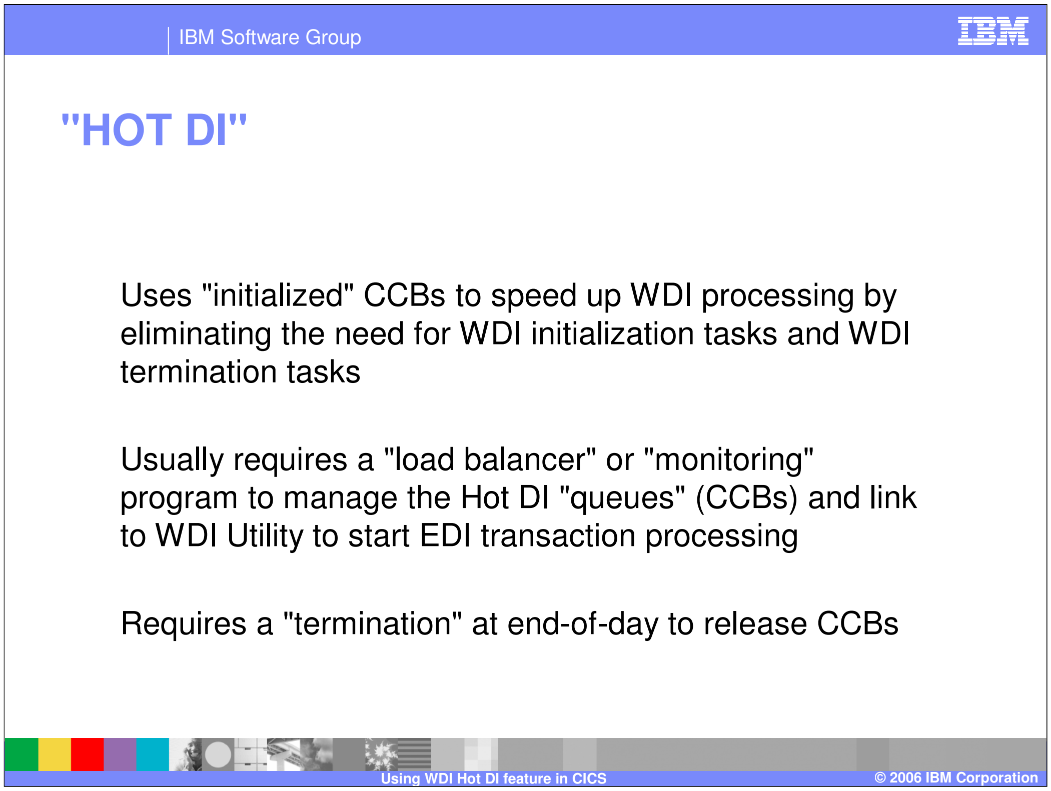


"HOT DI"

A performance technique available with WDI in CICS

Uses a Common Control Block CCB to store initialization values

“Hot DI” is a term used with WebSphere Data Interchange to describe a CICS environment whereby WDI processing “threads” are always initialized. Initialization occurs when WDI is getting starting. The product obtains control storage, determines some “instance options”, and prepares to process data. In a CICS environment there normally are many small transactions. By reducing the number of times WDI has to initialize itself, the time to process transactions after the first one can be significantly reduced. In this case WDI is always ready to process and already “warmed-up” \; hence the designation “Hot DI”.



"HOT DI"

Uses "initialized" CCBs to speed up WDI processing by eliminating the need for WDI initialization tasks and WDI termination tasks

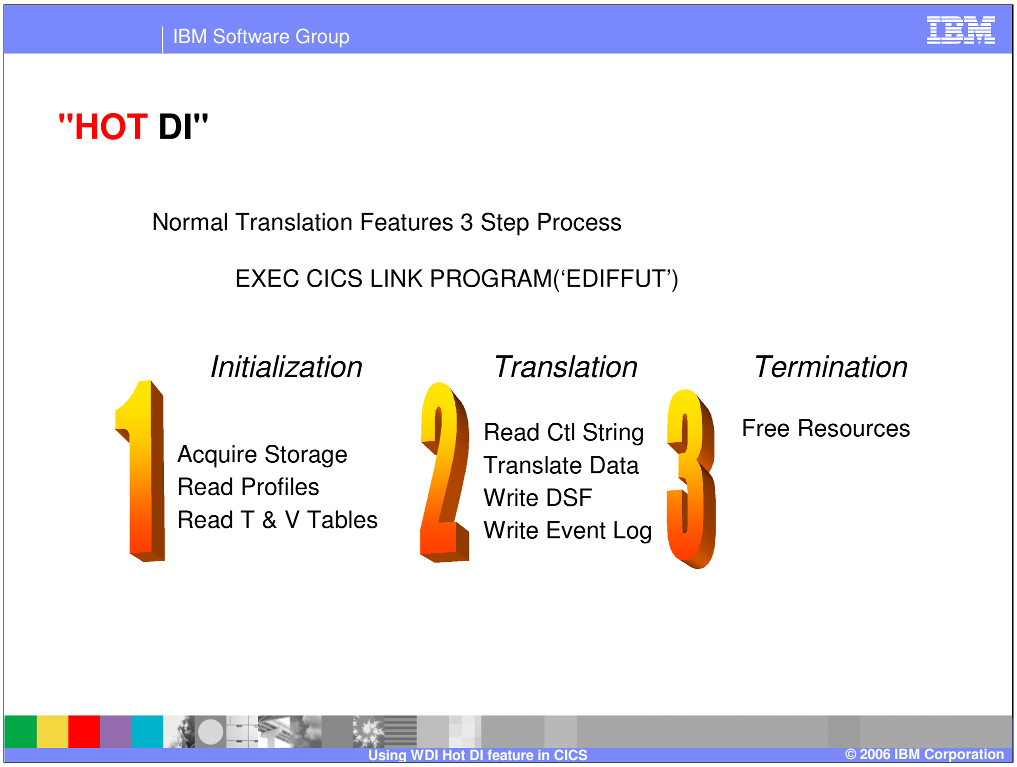
Usually requires a "load balancer" or "monitoring" program to manage the Hot DI "queues" (CCBs) and link to WDI Utility to start EDI transaction processing

Requires a "termination" at end-of-day to release CCBs

A normal invocation of WDI would have three phases: a) a one-time initiailization phase, b) a recurring translation phase, and c) a one-time termination phase. The translation phase would occur once for each transaction in a message, and for all messages in a file. In an on-line environment, each message is a “file” and thus WDI initializes and terminates may times.

With Hot DI, the initialized CCB is used in lieu of initalization and termination for each message.

The customer is required to provide a “load



"HOT DI"

Normal Translation Features 3 Step Process

EXEC CICS LINK PROGRAM('EDIFFUT')

1 *Initialization*

- Acquire Storage
- Read Profiles
- Read T & V Tables

2 *Translation*

- Read Ctl String
- Translate Data
- Write DSF
- Write Event Log

3 *Termination*

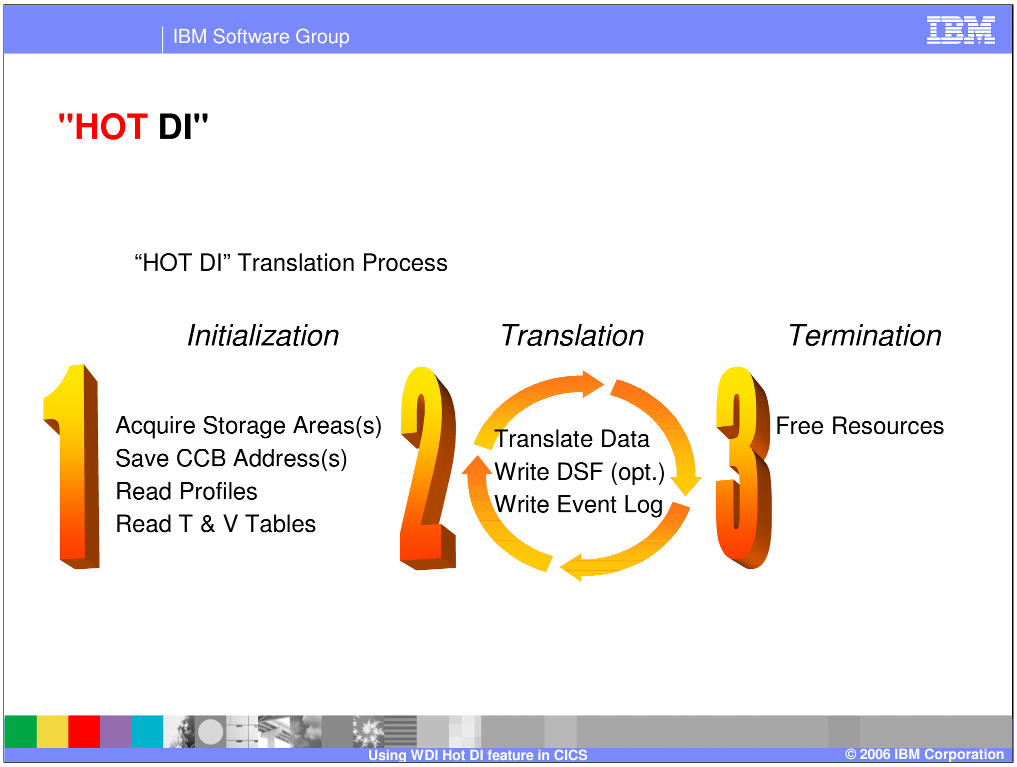
- Free Resources

This slide depicts the processing phases required by the load balancing program and the typical functions.

During an Initialization request WDI would acquire storage, possibly even read a control string, read some setup profiles, and read commonly used Translation and Validation tables,

During Translation, WDI would read a control string, perform the translation, write to Document Store and the Event Log.

At Termination, Database connections are released and storage is freed.



IBM Software Group

"HOT DI"

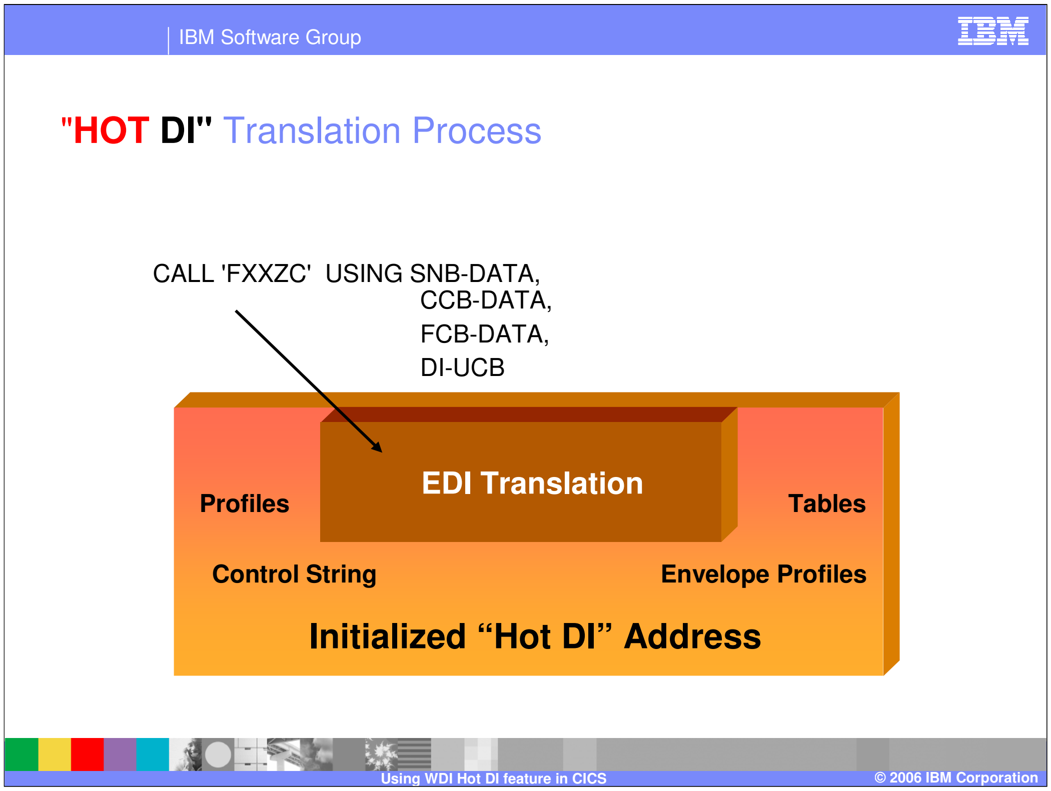
"HOT DI" Translation Process

Initialization	*Translation*	*Termination*
1	2	3
Acquire Storage Areas(s) Save CCB Address(s) Read Profiles Read T & V Tables	Translate Data Write DSF (opt.) Write Event Log	Free Resources

Using WDI Hot DI feature in CICS

© 2006 IBM Corporation

This shows why Hot DI can provide substantial performance improvement. The actions of phase 1 and phase 3 are only done once for the thread duration. If you had 100,000 discrete messages you would do 1 initialization, 100,000 translations, and 1 termination.



IBM Software Group

"HOT DI" Translation Process

```
CALL 'FXXZC'  USING SNB-DATA,
                    CCB-DATA,
                    FCB-DATA,
                    DI-UCB
```

Using WDI Hot DI feature in CICS

© 2006 IBM Corporation

Once the Hot-DI session is initialized, then calls to the session look like this.

The CCB contains the address of initialized CCB

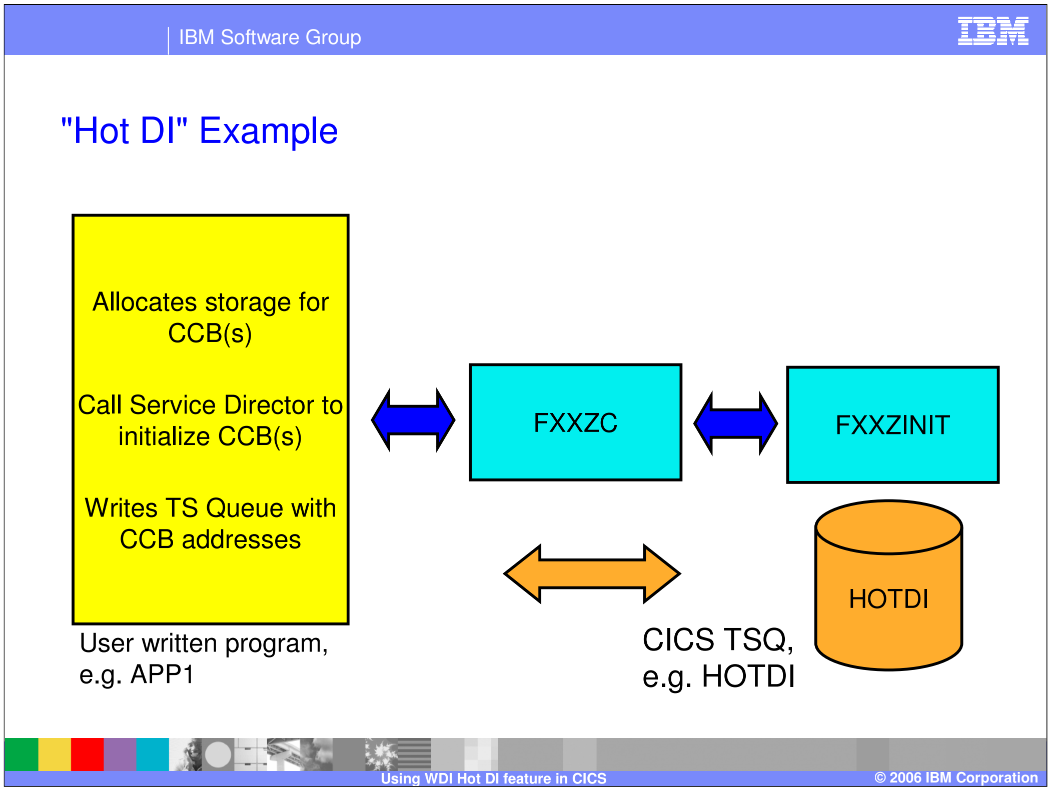
The Service Name Block (SNB) and Function Control Block (FCB) tell WDI what to do

The UCB contains the PERFORM COMMAND

FXXZC is the DI Service Director

interface to C programs

"Hot DI" Flow Example



IBM Software Group

"Hot DI" Example

Allocates storage for CCB(s)

Call Service Director to initialize CCB(s)

Writes TS Queue with CCB addresses

User written program, e.g. APP1

FXXZC

FXXZINIT

HOTDI

CICS TSQ, e.g. HOTDI

Using WDI Hot DI feature in CICS

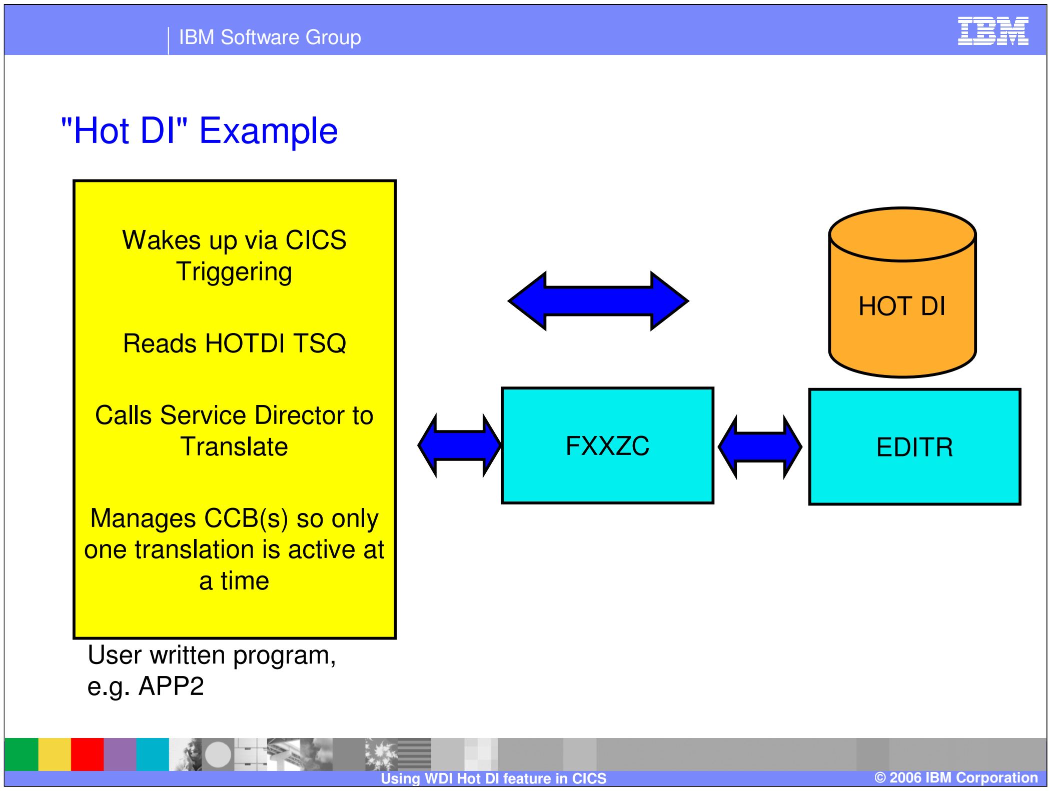
© 2006 IBM Corporation

The CCB is a WDI control block that contains information needed for subsequent processing.

In Hot DI, the user written, load balancing program (APP1) would acquire system storage and call WDI with an iniitializatoin request – passing an area that can be used to format a CCB. WDI would place values in the CCB and the User program would save the initialized CCB in a TSQ - like we would a COMMAREA in other CICS applications

FXXZC is WDI's program for interfacing with the WDI Service Director

FXZZINIT in the program which initializes WDI and the CCB



IBM Software Group

IBM

"Hot DI" Example

Wakes up via CICS Triggering

Reads HOTDI TSQ

Calls Service Director to Translate

Manages CCB(s) so only one translation is active at a time

User written program, e.g. APP2

FXXZC

EDITR

HOT DI

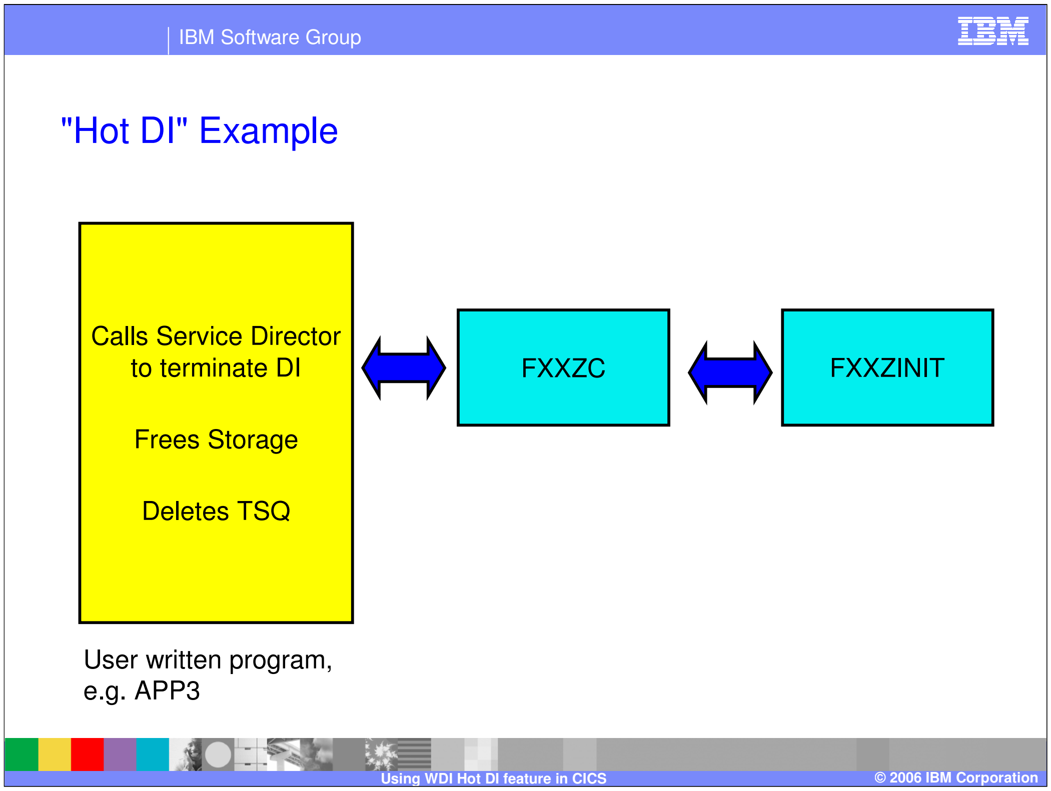
Using WDI Hot DI feature in CICS

© 2006 IBM Corporation

In this example, a second user program (APP2) is charged with getting the input message, and requesting translation. Optionally, it may pass the results to another application.

The APP2 program might be triggered by CICS or WMQ, then it would read a TSQ to get the initialized CCB. The WDI translator could then be invoked via a CALL. When translation is complete, control is returned to the calling program. APP2 is responsible for making sure only one translation request is active for a CCB at a time.

EDITR is the name of the WDI translator



"Hot DI" Example

Calls Service Director to terminate DI

Frees Storage

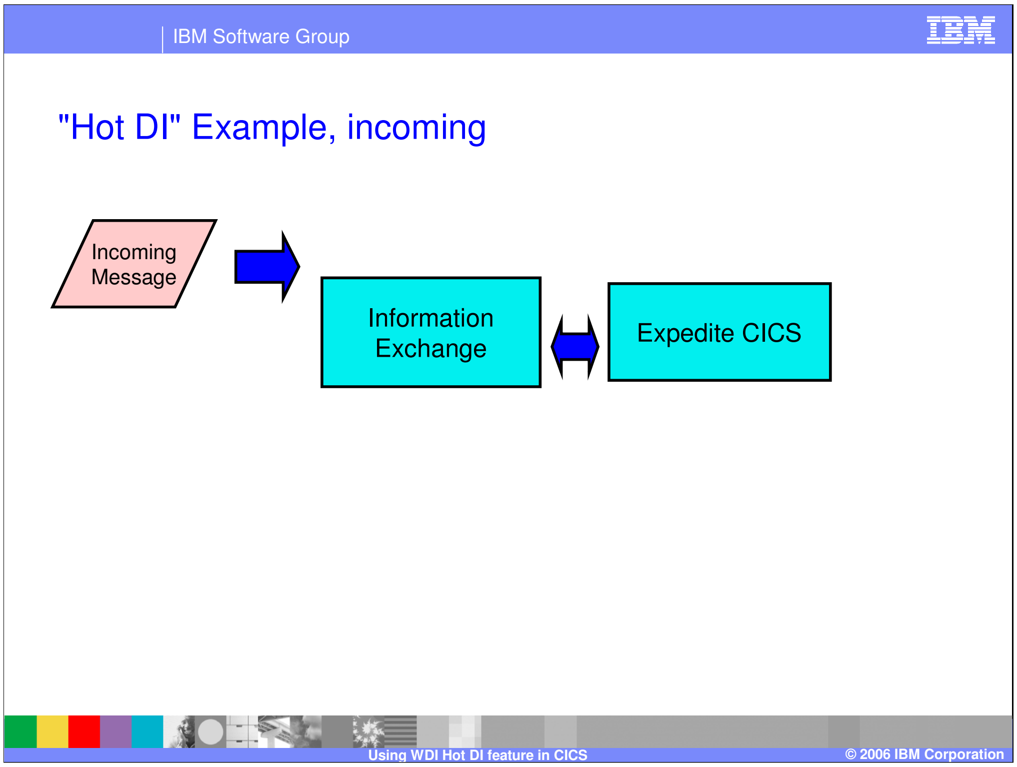
Deletes TSQ

FXXZC

FXXZINIT

User written program,
e.g. APP3

To complete this example, a third user program (APP3) cleans up storage by asking WDI to terminate and free up the CCB. It could be a second code path in the initialization program.



"Hot DI" Example, incoming

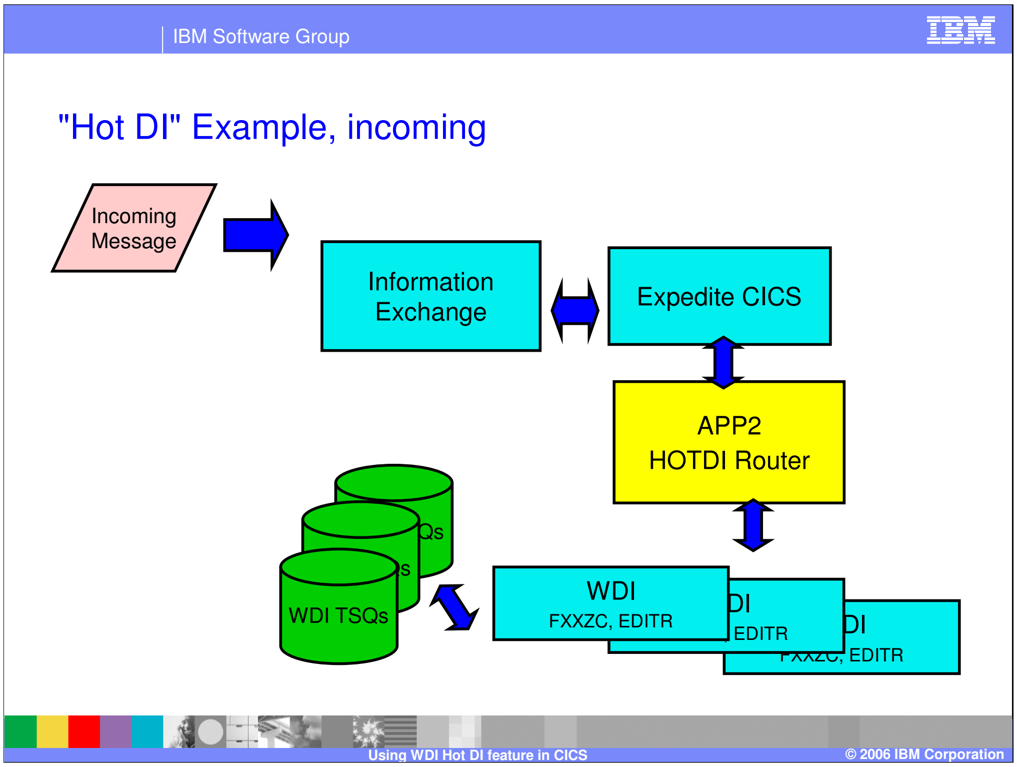
Incoming Message

Information Exchange

Expedite CICS

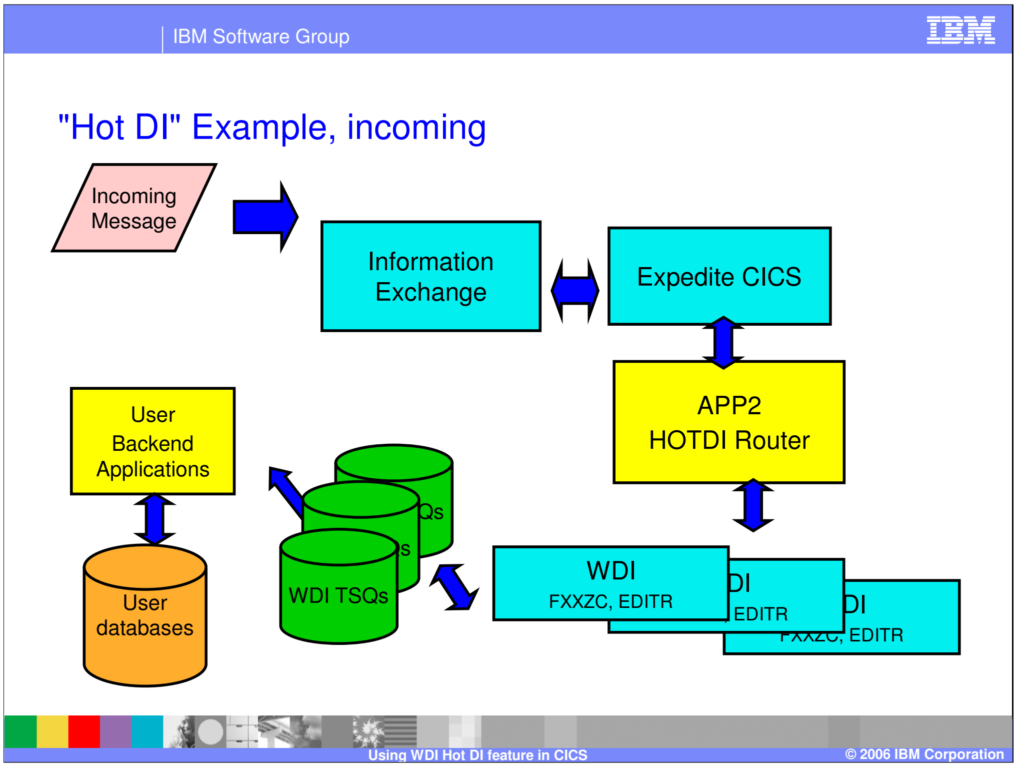
The data flow for an incoming message would be as follows:

1) The message would be placed in an Information Exchange (IE) mailbox by a trading partner.
2) Expedite CICS would be invoked to extract the message.



"Hot DI" Example, incoming

3) The user written load balancing program (APP2 in the prior explanation) would then invoke WDI to translate the message. TSQs would be used to house the initialized CCBs. A separate set of work TSQs would be required for each thread..



IBM Software Group

"Hot DI" Example, incoming

Incoming Message

Information Exchange

Expedite CICS

APP2 HOTDI Router

User Backend Applications

WDI TSQs

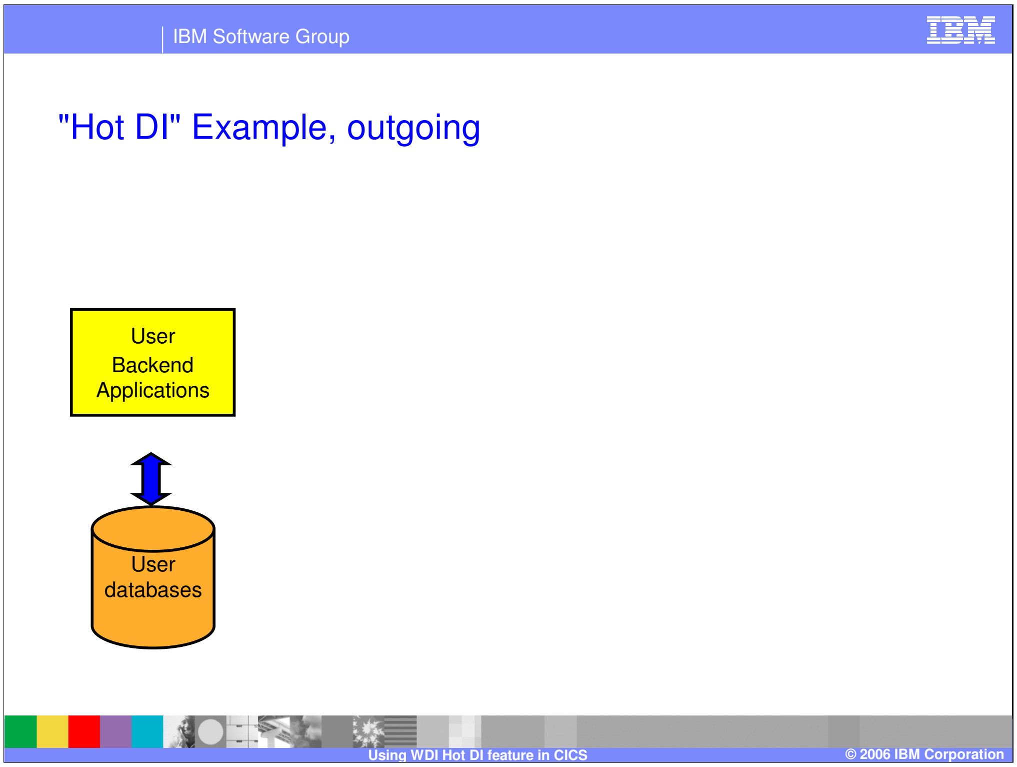
User databases

WDI FXXZC, EDITR

Using WDI Hot DI feature in CICS

© 2006 IBM Corporation

4) Backend applications would extract the translated data from the predetermined WDI work TSQs. User databases might be used to edit or store the data for the backend applications.



IBM Software Group

"Hot DI" Example, outgoing

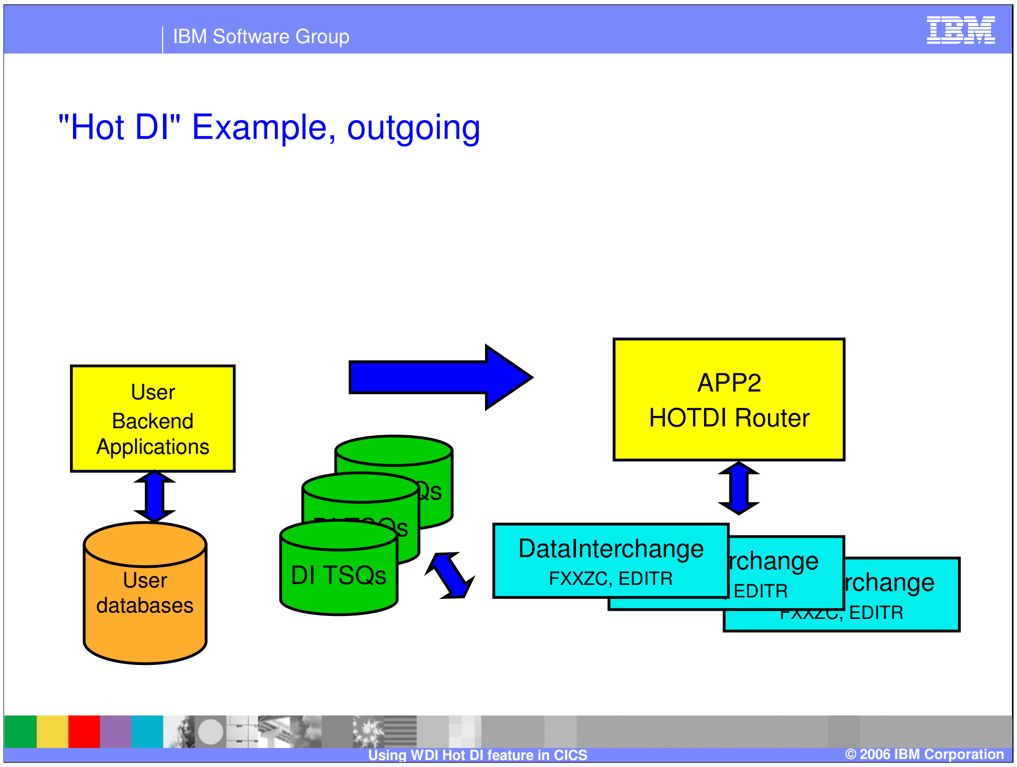
User Backend Applications

User databases

Using WDI Hot DI feature in CICS

An example or an outgoing message reverses the flow.

1) A user backend application would create data that needed to be translated.



IBM Software Group

IBM

"Hot DI" Example, outgoing

User Backend Applications

User databases

Qs

DI TSQs

DI TSQs

APP2
HOTDI Router

DataInterchange
FXXZC, EDITR

rchange
EDITR

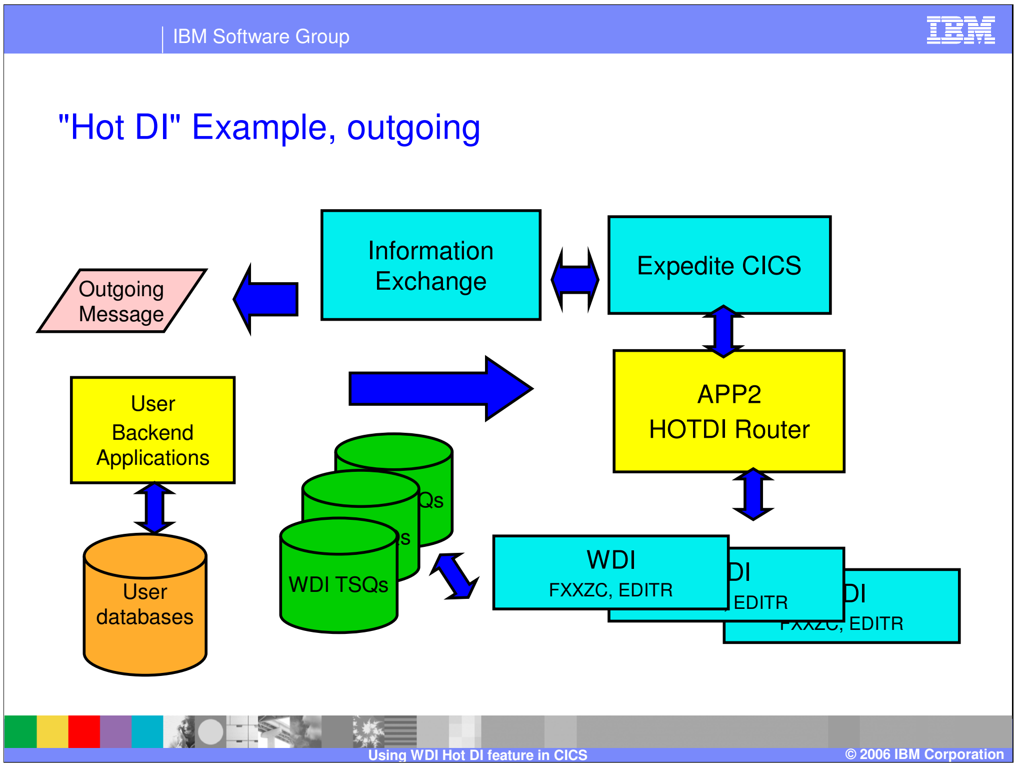
rchange
FXXZC, EDITR

Using WDI Hot DI feature in CICS

© 2006 IBM Corporation

2) Data would be placed on a TSQ and the user load balancing program (APP2) would be invoked.

3) APP2 would CALL WDI to translate the date



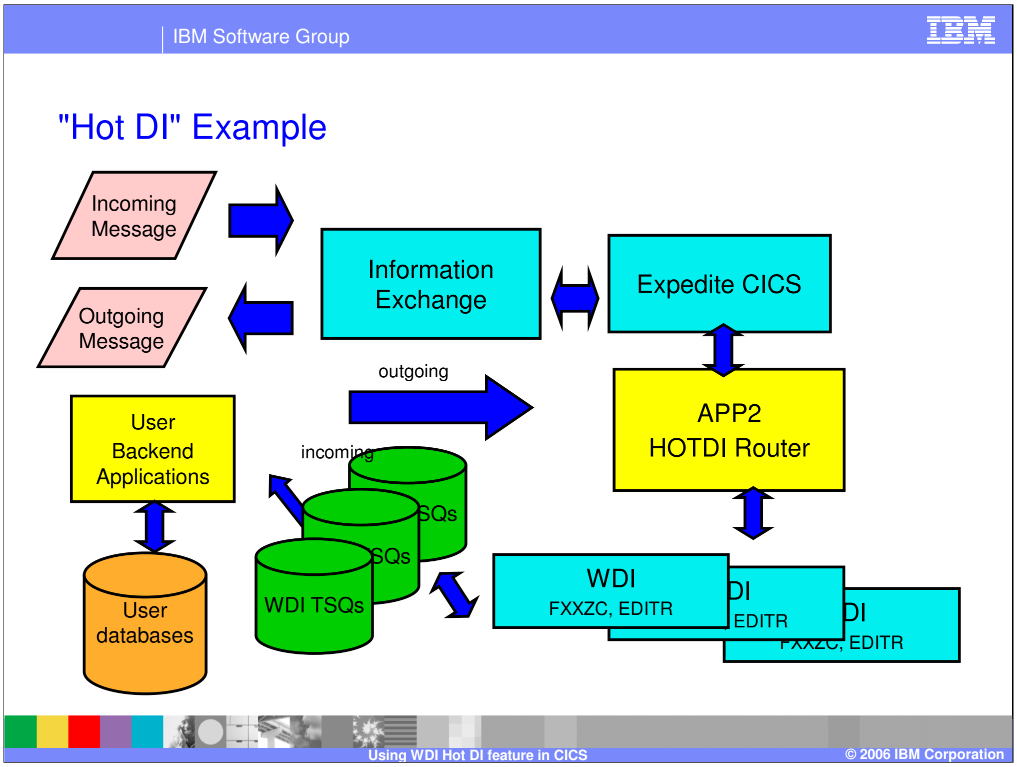
IBM Software Group

"Hot DI" Example, outgoing

Using WDI Hot DI feature in CICS

© 2006 IBM Corporation

4) The data would be passed to Expedite CICS and sent to IE



IBM Software Group

"Hot DI" Example

Incoming Message

Outgoing Message

Information Exchange

Expedite CICS

outgoing

incoming

APP2 HOTDI Router

User Backend Applications

User databases

WDI TSQs

SQs

SQs

WDI FXXZC, EDITR

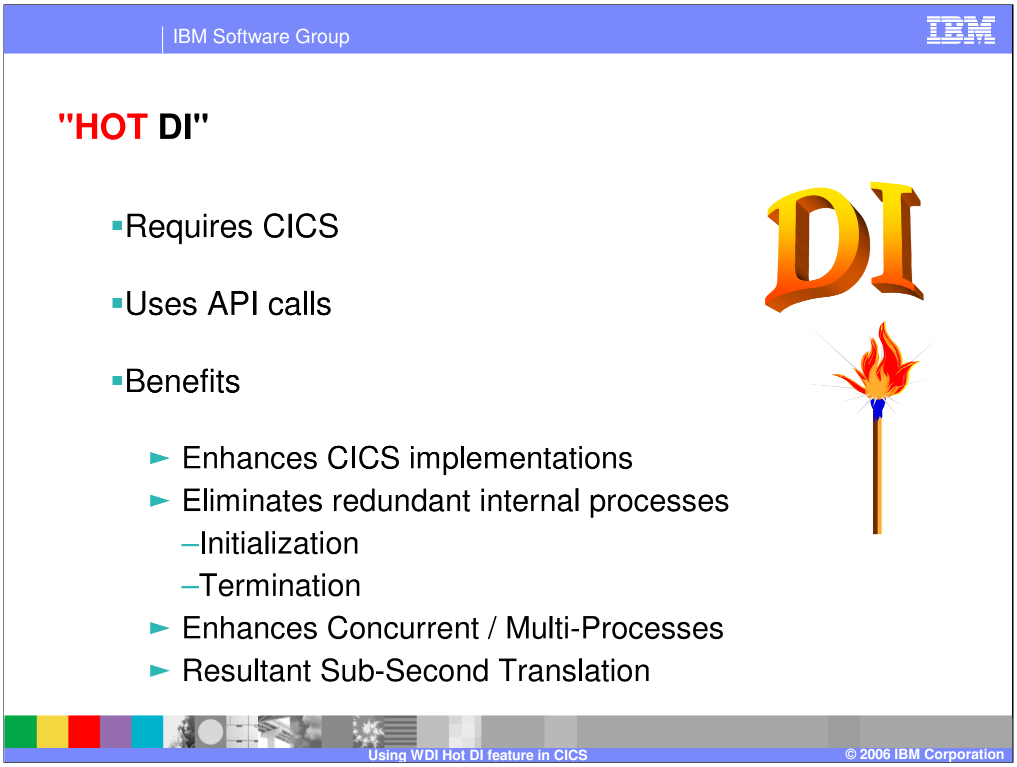
DI EDITR

DI FXXZC, EDITR

Using WDI Hot DI feature in CICS

© 2006 IBM Corporation

A review of the complete flow.



IBM Software Group

"HOT DI"

- Requires CICS
- Uses API calls
- Benefits
  - ► Enhances CICS implementations
  - ► Eliminates redundant internal processes
    - –Initialization
    - –Termination
  - ► Enhances Concurrent / Multi-Processes
  - ► Resultant Sub-Second Translation

Using WDI Hot DI feature in CICS

© 2006 IBM Corporation

A review of Hot DI

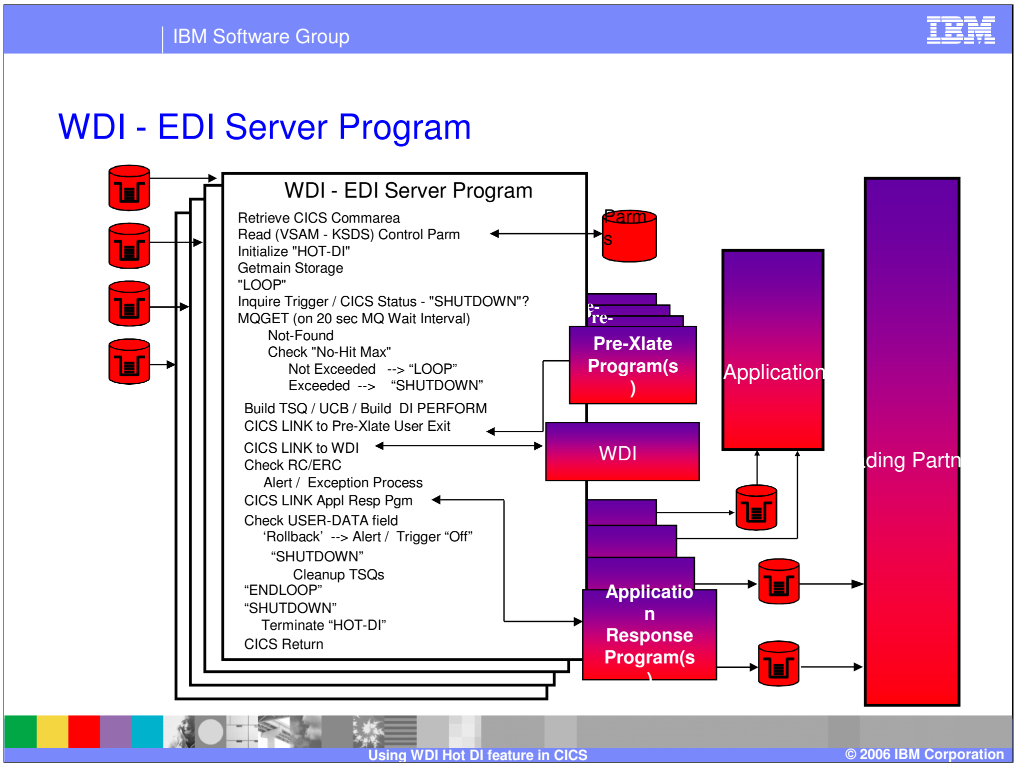
1) It is used exlusively in CICS

2) It uses API calls to initialize, translate, and terminate WDI

3) Its benefits included faster response time, with less WDI processing, and the ability to multi-thread from a single data source

IBM Software Group

WDI - EDI Server Program



```
Retrieve CICS Commarea
Read (VSAM - KSDS) Control Parm
Initialize "HOT-DI"
Getmain Storage
"LOOP"
Inquire Trigger / CICS Status - "SHUTDOWN"?
MQGET (on 20 sec MQ Wait Interval)
     Not-Found
     Check "No-Hit Max"
         Not Exceeded   --> "LOOP"
         Exceeded  -->    "SHUTDOWN"
 Build TSQ / UCB / Build  DI PERFORM
 CICS LINK to Pre-Xlate User Exit
 CICS LINK to WDI
 Check RC/ERC
     Alert /  Exception Process
 CICS LINK Appl Resp Pgm
 Check USER-DATA field
     'Rollback'  --> Alert /  Trigger "Off"
      "SHUTDOWN"
          Cleanup TSQs
 "ENDLOOP"
 "SHUTDOWN"
     Terminate "HOT-DI"
 CICS Return
```

Using WDI Hot DI feature in CICS

© 2006 IBM Corporation

In summary,

A typical user written load balancing program would be as follows:

Retrieve CICS Commarea
Read (VSAM - KSDS) Control Parm
Initialize "HOT-DI"
Getmain Storage
"LOOP"
Inquire Trigger / CICS Status - "SHUTDOWN"?
MQGET (on 20 sec MQ Wait Interval)
Not-Found
Check "No-Hit Max“
Not Exceeded --> “LOOP”
Exceeded --> “SHUTDOWN”
Build TSQ / UCB / Build DI PERFORM
CICS LINK to Pre-Xlate User Exit
CICS LINK to WDI
Check RC/ERC
Alert / Exception Process
CICS LINK Appl Resp Pgm
Check USER-DATA field
‘Rollback’ --> Alert / Trigger “Off”
“SHUTDOWN”
Cleanup TSQs
“ENDLOOP”
“SHUTDOWN”
Terminate “HOT-DI”
CICS Return

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.