

---

# Ingest Data Base Design Review

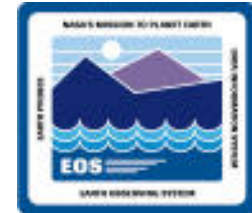
**Bryn Ardanuy**

[bardanuy@eos.hitc.com](mailto:bardanuy@eos.hitc.com)

---

**15 April 1996**

# Ingest Data Base Overview

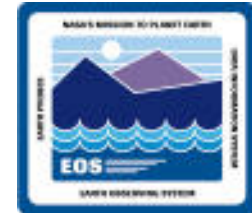


## Data base access

### Data base tables

- Checkpoint tables
- Operator interface tables
- Data preprocessing tables
  - Data type insertion procedure

# Ingest Data Base Access



## **DBAccess class**

- **Used to connect to and disconnect from the data base**

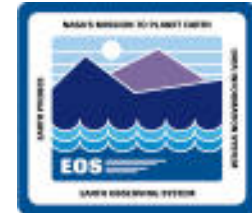
## **Individual data base table classes**

- **Used to insert, update and delete**

## **Data base access is done via:**

- **SYBASE Open Client Client-Library**
- **Stored procedures**
- **SQL commands**

# Ingest Data Base Tables



## Checkpoint Tables

- InRequestProcessHeader
- InRequestProcessData
- InRequestFileInfo
- InSessionInfo
- InValRequestState
- InValIngestType
- InValDataGranuleState

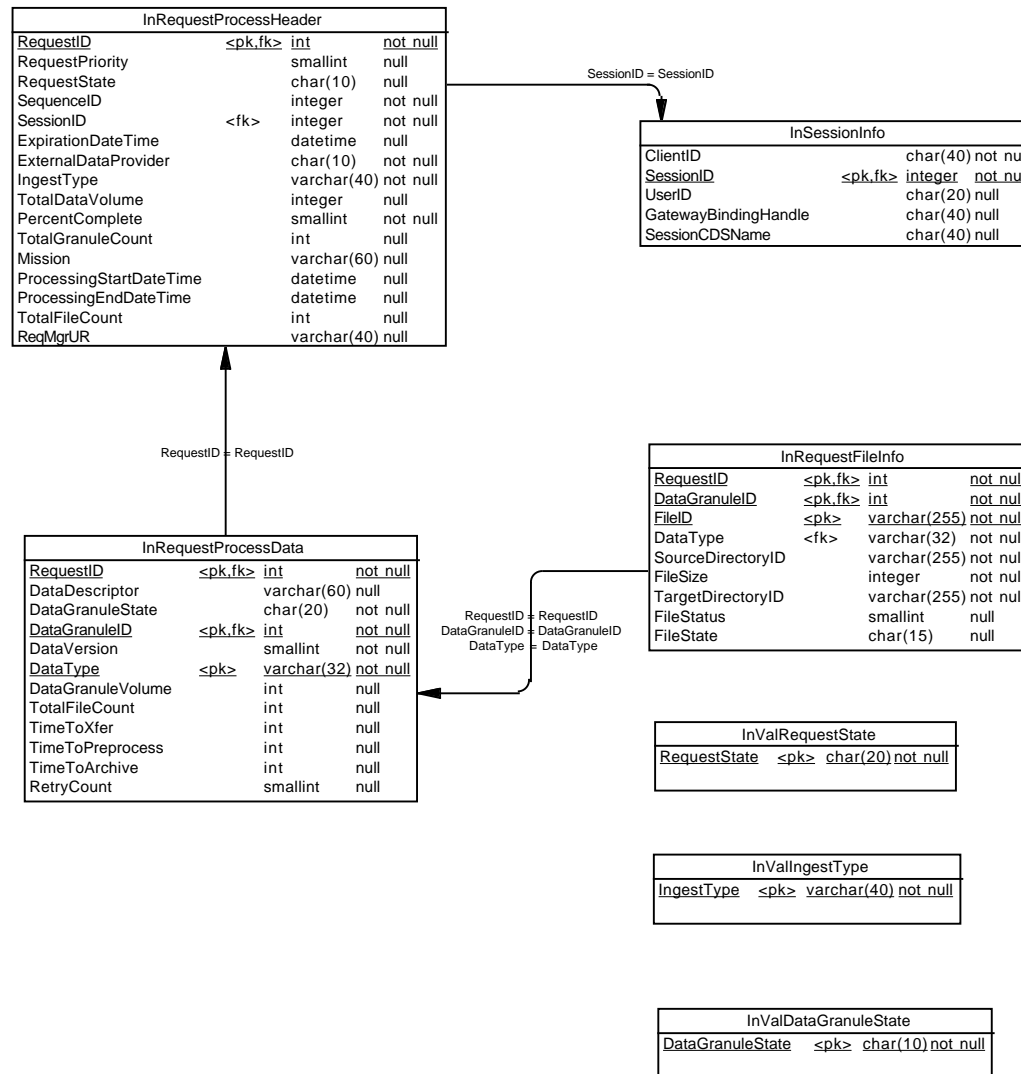
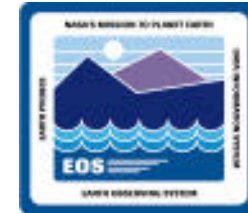
## Operator Interface Tables

- Monitor and Control
  - InRequestProcessHeader
  - InRequestProcessData
  - InMediaCheckin
- History Log
  - InRequestSummaryHeader
  - InRequestSummaryData
- Pulldown Menus
  - InExternalDataProviderInfo
  - InDataTypeTemplate

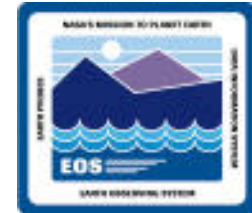
## Data Preprocessing Tables (Templates)

- InDataTypeTemplate
- InFileTypeTemplate
- InSourceMCF

# Checkpoint and Operator Interface Tables



# Checkpoint and Operator Interface Tables



InExternalDataProviderInfo			
ExternalDataProvider	<pk>	char(10)	not null
IngestType		varchar(40)	not null

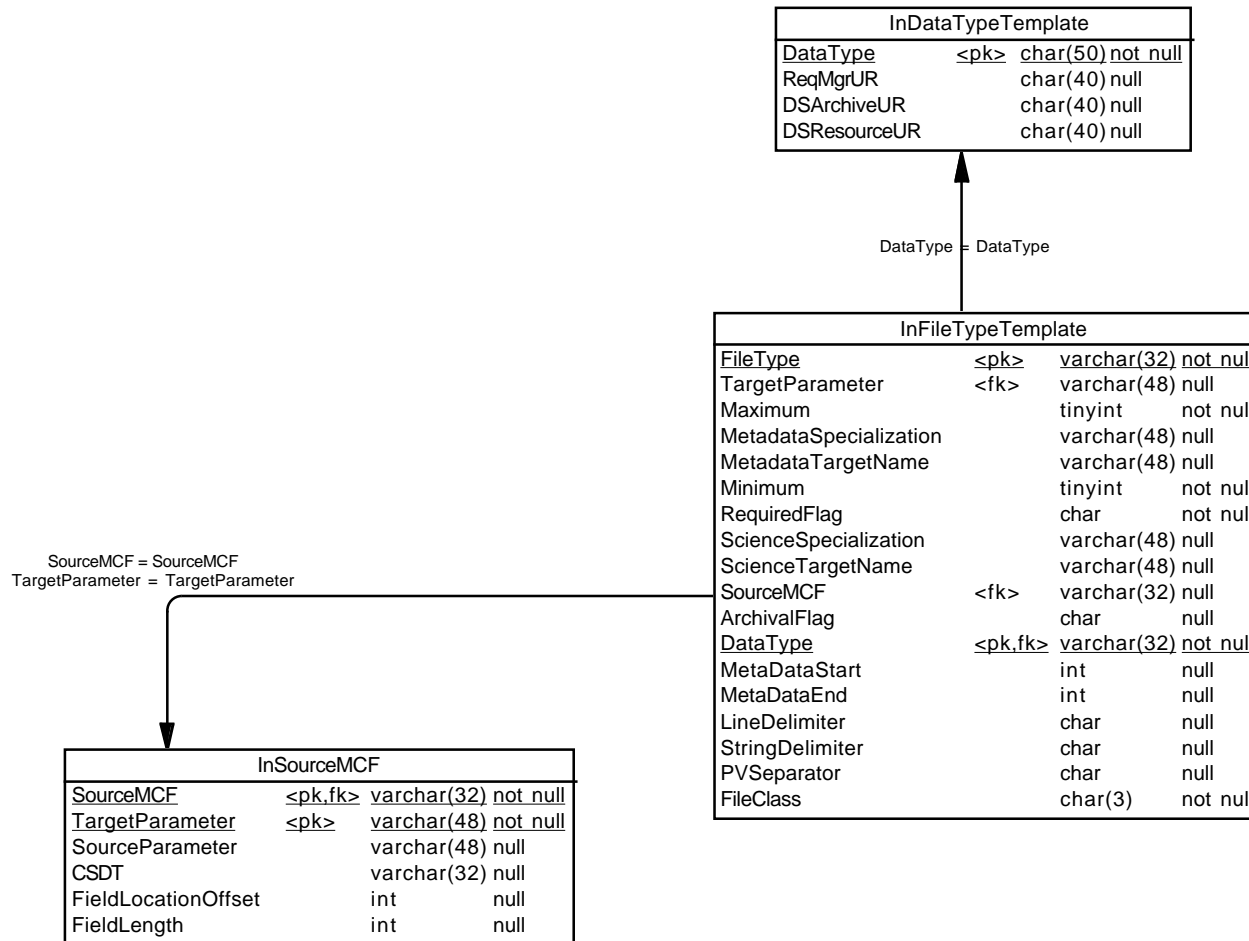
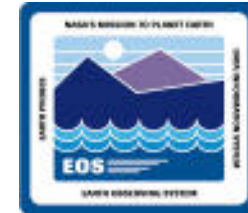
InMediaCheckin			
VolumeID	<pk>	varchar(40)	not null
ExternalDataProvider		char(10)	null
CheckinTime		datetime	not null
Status		char(10)	null
State		char(15)	not null
CompletionTime		datetime	null

InRequestSummaryHeader			
ExternalDataProvider		char(10)	not null
Mission		varchar(60)	null
RequestID	<pk, fk>	int	not null
ProcessingEndDateTime		datetime	null
ProcessingStartDateTime		datetime	null
TotalDataVolume		integer	null
TotalGranuleCount		integer	not null
TotalFileCount		integer	null
IngestType		varchar(40)	not null
TimeToXfer		int	null
TimeToPreprocess		int	null
TimeToArchive		int	null
TotalSuccessfulGranules		int	null

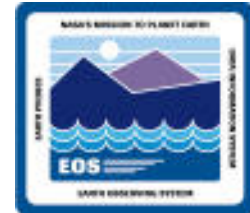
RequestID = RequestID

InRequestSummaryData			
DataType	<pk>	varchar(32)	not null
RequestID	<pk, fk>	int	not null
DataGranuleID	<pk>	int	not null
DataGranuleVolume		integer	null
DataGranuleState		int	null
TotalFileCount		int	null
TimeToPreprocess		int	null
TimeToArchive		int	null
TimeToXfer		int	null
RetryCount		smallint	null

# Data Preprocessing Tables



# Ingest Checkpoint Tables



**Contain detailed Ingest request status information**

**Used to restore system state after a failure**

**Used to validate data being inserted**



# Ingest Operator Interface Tables



**Displays status of ongoing requests**

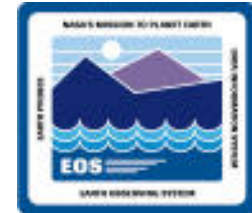
**Displays summary information for completed requests**

**Displays selections of External Data Providers and Data Types**

**Enters tape information**

**Displays status of tapes**

# Ingest Data Preprocessing Tables (Templates)



**Store template information to drive Ingest data:**

- Preprocessing
- Metadata extraction
- Validation

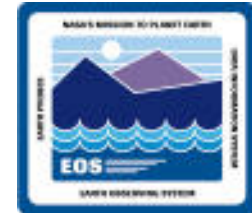
**Template types**

- Data Type
- File Type
- Metadata Configuration

**Insertion of data types allows for extensibility of the system generally without any software changes**

- Software changes required for new metadata extraction and conversion algorithms and for new data conversion and reformatting algorithms

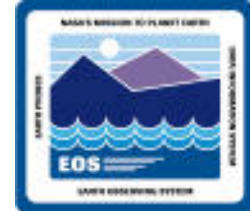
# Data Type Insertion Procedure



## 3 Step Process:

- Add data type preprocessing information to the test data base
- Validate the test data base
- Move the validated template tables to the operational data base

# Data Type Insertion Procedure - Add Data Type To Data Base



**Edit sample batch files for stored procedures:**

- **AddANewDataType**
- **AddANewFileTypeRow**
- **AddANewSourceMCFRow**

**Execute the batch files via Sybase ISQL in a test database**

- **isql -U username -P password -i batchfilename -o outputfile**
- **The batch files execute stored procedures which perform error checking on the input data**

# Batch File Example



```
USE IngestTst1 -- Change to appropriate database name
```

```
GO
```

```
DECLARE
```

```
    @source_MCF varchar(32), -- metadata configuration file for data type
```

```
    @target_parameter varchar(48), -- attribute name in DID 311 and Data  
                                   -- Server
```

```
    @source_parameter varchar(48), -- alias for target parameter in raw file
```

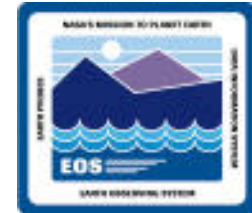
```
    @CSDT varchar(32), -- computer science data type (float, int, etc.)
```

```
    @field_location_offset integer, -- depending on file class, either byte  
                                   -- offset or location
```

```
    @field_length integer -- length of value information
```

(continued on next page)

# Batch File Example (cont.)



-- Change the values of the following parameters to desired values

```
SELECT @source_MCF = "AFIELD1",
```

```
    @target_parameter = "SOUTHBOUNDINGLATITUDE",
```

```
    @source_parameter = "a",
```

```
    @CSDT = "float",
```

```
    @field_location_offset = 4,
```

```
    @field_length = 4
```

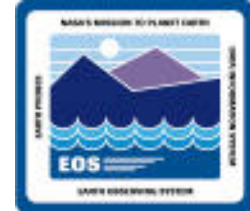
```
EXECUTE AddANewSourceMCFRow @source_MCF, @target_parameter,
```

```
    @source_parameter, @CSDT, @field_location_offset,
```

```
    @field_length
```

```
GO
```

# Data Type Insertion Procedure - Validate Inserted Data



## Validate the inserted data in the test data base

- This is done by Integration and Test personnel
  - Test the new data type
  - Regression test the remainder of the data base

# Data Type Insertion Procedure - Move Inserted Data To Operational Data Base



## Backup modified template tables

- Using the SQL bcp (bulk copy) command, write the data from the test data base tables into files

## Drop template tables from operational database

- Using the truncate table SQL command

## Insert the modified template tables into the operational database

- Using the SQL bcp command, write the data from the backup files into the operational database



# Summary



## The Ingest data base provides:

- Ability to recover after a failure
- Flexibility and extensibility
- Storage for ongoing request and history data
- Data validation