
CHIPS Utility Service

The CHIPS Utility service is responsible for the start of day, end of day, and CHIPS adapter lookup functions.

The following table provides an overview of the CHIPS Utility service:

System Name	CHIPSUtilityservice
Graphical Process Modeler (GPM) categories	All Services.
Description	This service is responsible for the start of day, end of day, and CHIPS adapter lookup functions.
Business usage	This service is used by the CHIPS adapter to execute: <ul style="list-style-type: none">◆ Start of Day functions (reset the CHIPS status to 1, CHIPS message number to 1, and the MQ_counter to 0)◆ End of Day functions (move the outbound messages into the history table, move the inbound messages into the history table, update Mailbox_id in the MBX table with the Mailbox_id of the <participant_num>_history mailbox)◆ CHIPS adapter lookups (based on the channel information, get the CHIPS adapter name related to the appropriate set)
Usage example	Used to prepare for daily CHIPS usage, perform end of day housekeeping functions to properly store the data in the database, and look up the CHIPS adapter name when a message is received from the WebsphereMQ Async Receiver adapter (to allow the application to send back an acknowledgement message using the correct CHIPS adapter and transport mode). The predefined CHIPSUtility_ReceiveHandler business process is provided to handle incoming messages from CHIPS.
Preconfigured?	Yes.
Requires third party files?	No.
Platform availability	All supported application platforms.
Related services	This service works with the CHIPS adapter.
Application requirements	None.
Initiates business processes?	No. However, a predefined business process (CHIPSUtilitySOD) is provided to execute the CHIPS Utility service start-of-day actions. You can schedule this predefined business process to be started every morning. Also, a predefined business process (CHIPSUtilityEOD) is provided to execute the CHIPS Utility service end-of-day actions. You can schedule this predefined business process to be started every evening.
Invocation	A user who has permission to perform this activity must execute the business process that invokes this service.
Business process context considerations	N/A

Returned status values	Success, Failure.
Restrictions	None.
Persistence level	N/A
Testing considerations	To test this service, run the CHIPS Utility service business process and verify that it completes successfully. Debug information for this service is located at: Operations > System > Logs > CHIPS

How the CHIPS Utility Service Works

The CHIPS Utility service prepares for daily CHIPS usage, performs end of day housekeeping functions to properly store the data in the database, and perform lookups to the CHIPS adapter name when a message is received from the WebsphereMQ Async Receiver adapter (to allow the application to send back an acknowledgement message using the correct CHIPS adapter and transport mode).

Implementing the CHIPS Utility Service

You do not need to do anything to implement the CHIPS Utility service.

Configuring the CHIPS Utility Service

You do not need to do anything to configure the CHIPS Utility service.

Parameters Passed From Business Process to Service

The following table contains the parameter passed from the business process to the CHIPS Utility service:

Parameter	Description
serviceName	This is the CHIPS Adapter Instance Name.

Business Process Example

This example uses the following message types for CHIPS Utility service:

- ◆ startOfDay
- ◆ endOfDay
- ◆ handleCHIPSReceiveRequest

This example sets the start of day parameters:

```
<process name="CHIPSUtilitySOD">
  <sequence name="CHIPSUtility_StartOfDay">
    <operation name="set user token">
```

```

    <participant name="SetUserToken"/>
    <output message="SetUserTokenMessage">
      <assign to="USER_TOKEN">admin</assign>
      <assign to="." from="*" />
    </output>
    <input message="inmsg">
      <assign to="." from="*" />
    </input>
  </operation>
<!-- Start of Day Process-->
<operation>
  <participant name="CHIPSUtilityService"/>
  <output message="startOfDay">
    <assign to="." from="*" />
    <assign to="serviceName">CHIPSAdapter</assign>
  </output>
  <input message="testing">
    <assign to="." from="*" />
  </input>
</operation>
</sequence>
</process>

```

This example sets the end of day parameters:

```

<process name="CHIPSUtilityEOD">
  <sequence name="CHIPSUtility_EndOfDay">
    <operation name="set user token">
      <participant name="SetUserToken"/>
      <output message="SetUserTokenMessage">
        <assign to="USER_TOKEN">admin</assign>
        <assign to="." from="*" />
      </output>
      <input message="inmsg">
        <assign to="." from="*" />
      </input>
    </operation>
  <!-- End of Day Process-->
  <operation>
    <participant name="CHIPSUtilityService"/>
    <output message="endOfDay">
      <assign to="." from="*" />
      <assign to="serviceName">CHIPSAdapter</assign>
    </output>
    <input message="testing">
      <assign to="." from="*" />
    </input>
  </operation>
</sequence>
</process>

```

This example handles incoming CHIPS requests. The CHIPSUtility_ReceiveHandler is preloaded into the application:

```

<process name="CHIPSUtility_ReceiveHandler">
  <sequence name="CHIPSAdapter">

```

```
<operation name="set user token">
  <participant name="SetUserToken"/>
  <output message="SetUserTokenMessage">
    <assign to="USER_TOKEN">admin</assign>
    <assign to="." from="*" />
  </output>
  <input message="inmsg">
    <assign to="." from="*" />
  </input>
</operation>
<!-- handle incoming CHIPS request -->
<operation>
  <participant name="CHIPSUtilityService"/>
  <output message="handleCHIPSReceiveRequest">
    <assign to="." from="*" />
  </output>
  <input message="testing">
    <assign to="." from="*" />
  </input>
</operation>
</sequence>
</process>
```